

Package ‘flexible’

January 25, 2021

Type Package

Title Functions for Tabular Reporting

Version 0.6.2

Description Create pretty tables for 'HTML', 'PDF', 'Microsoft Word' and 'Microsoft PowerPoint' documents from 'R Markdown'. Functions are provided to let users create tables, modify and format their content. It also extends package 'officer' that does not contain any feature for customized tabular reporting.

License GPL-3

LazyData TRUE

Imports stats, utils, grDevices, graphics, officer (>= 0.3.15),
rmarkdown, knitr, htmltools, xml2, data.table (>= 1.13.0), uuid
(>= 0.1-4), gdttools (>= 0.1.6), rlang, base64enc

RoxygenNote 7.1.1

Suggests testthat (>= 2.1.0), xtable, webshot, magick, ggplot2,
scales, broom, mgcv

Encoding UTF-8

URL <https://ardata-fr.github.io/flexible-book/>,
<https://davidgohel.github.io/flexible/>

BugReports <https://github.com/davidgohel/flexible/issues>

VignetteBuilder knitr

NeedsCompilation no

Author David Gohel [aut, cre],
Quentin Fazilleau [ctb],
Maxim Nazarov [ctb] (rmarkdown for docx output),
Titouan Robert [ctb],
Michael Barrowman [ctb] (inline footnotes),
Atsushi Yasumoto [ctb] (support for bookdown cross reference),
Paul Julian [ctb] (support for gam objects)

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

Repository CRAN

Date/Publication 2021-01-25 13:40:06 UTC

R topics documented:

flextable-package	4
add_body	5
add_header	6
add_header_lines	7
add_header_row	8
align	9
as_b	10
as_bracket	11
as_chunk	12
as_flextable	13
as_flextable.gam	13
as_flextable.glm	14
as_flextable.grouped_data	15
as_flextable.htest	16
as_flextable.lm	17
as_flextable.xtable	17
as_grouped_data	19
as_i	20
as_image	21
as_paragraph	22
as_raster	23
as_sub	24
as_sup	25
autofit	26
bg	27
body_add_flextable	28
bold	29
border_inner	30
border_inner_h	31
border_inner_v	32
border_outer	33
border_remove	34
colformat_char	34
colformat_date	35
colformat_datetime	36
colformat_double	37
colformat_image	38
colformat_int	39
colformat_lgl	40
colformat_num	41
color	42
compose	44
continuous_summary	45
delete_part	46
dim.flextable	46
dim_pretty	47

empty_blanks	48
fit_to_width	49
fix_border_issues	50
flextable	50
flextable_dim	52
flextable_html_dependency	52
flextable_to_rmd	53
font	54
fontsize	55
footers_flextable_at_bkm	56
footnote	56
fp_border_default	58
fp_text_default	58
get_flextable_defaults	59
gg_chunk	60
headers_flextable_at_bkm	61
height	62
highlight	63
hline	64
hline_bottom	65
hline_top	66
hrule	67
htmltools_value	68
hyperlink_text	68
italic	69
knit_print.flextable	70
linerange	73
line_spacing	75
lollipop	75
merge_at	77
merge_h	78
merge_h_range	78
merge_none	79
merge_v	80
minibar	81
padding	82
ph_with.flextable	84
plot.flextable	85
plot_chunk	86
print.flextable	87
proc_freq	88
rotate	89
save_as_docx	90
save_as_html	91
save_as_image	92
save_as_pptx	93
set_caption	94
set_flextable_defaults	95

set_formatter	97
set_header_footer_df	98
set_header_labels	99
set_table_properties	100
style	101
theme_alafoli	102
theme_booktabs	103
theme_box	104
theme_tron	104
theme_tron_legacy	105
theme_vader	106
theme_vanilla	106
theme_zebra	107
valign	108
vline	109
vline_left	110
vline_right	111
void	112
width	112

Index	114
--------------	------------

flextable-package *flextable: Functions for Tabular Reporting*

Description

The flextable package facilitates access to and manipulation of tabular reporting elements from R.

The documentation of functions can be opened with command `help(package = "flextable")`.

To learn more about flextable, start with the vignettes: `browseVignettes(package = "flextable")`.

`flextable()` function is producing flexible tables where each cell can contain several chunks of text with their own set of formatting properties (bold, font color, etc.). Function `compose()` lets customise text of cells.

See Also

<https://davidgohel.github.io/flextable/>, `flextable()`

add_body	<i>Add rows in body part</i>
----------	------------------------------

Description

Add rows in the flextable's body. It can be inserted at the top or the bottom. The function is column oriented, labels are specified for each columns, there can be more than a value - resulting in more than a new row.

Usage

```
add_body(x, top = TRUE, ..., values = NULL)
```

Arguments

x	a flextable object
top	should the rows be inserted at the top or the bottom.
...	a named list (names are data colnames) of strings specifying corresponding values to add. It is important to insert data of the same type as the original data, otherwise it will be transformed (probably into strings if you add a character' where a double' is expected). This keeps the ability to format cell contents with the colformat_* functions, for example <code>colformat_num()</code> .
values	a list of name-value pairs of labels or values, names should be existing col_key values. This argument can be used instead of ... for programming purpose (If values is supplied argument ... is ignored).

See Also

[flextable\(\)](#), [add_header\(\)](#), [add_footer\(\)](#)

Examples

```
ft <- flextable(head(iris),
  col_keys = c(
    "Species", "Sepal.Length", "Petal.Length",
    "Sepal.Width", "Petal.Width"
  )
)

ft <- add_body(
  x = ft, Sepal.Length = 1:5,
  Sepal.Width = 1:5 * 2, Petal.Length = 1:5 * 3,
  Petal.Width = 1:5 + 10, Species = "Blah", top = FALSE
)

ft <- theme_booktabs(ft)
ft
```

add_header	<i>Add a rows of labels in header or footer part</i>
------------	--

Description

Add rows of labels in the flextable's header or footer part. It can be inserted at the top or the bottom of the part. The function is column oriented, labels are specified for each columns, there can be more than a label - resulting in more than a new row.

Usage

```
add_header(x, top = TRUE, ..., values = NULL)
```

```
add_footer(x, top = TRUE, ..., values = NULL)
```

Arguments

x	a flextable object
top	should the rows be inserted at the top or the bottom.
...	a named list (names are data colnames) of strings specifying corresponding values to add. It is important to insert data of the same type as the original data, otherwise it will be transformed (probably into strings if you add a character' where a double' is expected). This keeps the ability to format cell contents with the colformat_* functions, for example colformat_num() .
values	a list of name-value pairs of labels or values, names should be existing col_key values. This argument can be used instead of ... for programming purpose (If values is supplied argument ... is ignored).

Illustrations

Note

when repeating values, they can be merged together with function [merge_h\(\)](#) and [merge_v\(\)](#).

See Also

Other headers and footers: [add_header_lines\(\)](#), [add_header_row\(\)](#), [set_header_footer_df](#), [set_header_labels\(\)](#)

Examples

```
ft <- flextable( head( iris ),
  col_keys = c("Species", "Sepal.Length", "Petal.Length",
    "Sepal.Width", "Petal.Width") )

# start with no header
ft <- delete_part(ft, part = "header")

# add a line of row
ft <- add_header(x = ft, Sepal.Length = "length",
  Sepal.Width = "width", Petal.Length = "length",
  Petal.Width = "width", Species = "Species", top = FALSE )
# add another line of row at the top position
ft <- add_header(ft, Sepal.Length = "Inches",
  Sepal.Width = "Inches", Petal.Length = "Inches",
  Petal.Width = "Inches", top = TRUE )
# merge horizontally when there are identical values
ft <- merge_h(ft, part = "header")

# add a footnote in the footer part
ft <- add_footer(ft, Species = "This is a note in footer" )
ft <- merge_at(ft, j = 1:5, part = "footer")

# theme the table
ft <- theme_box(ft)

ft
```

add_header_lines *Add a label in a header or footer new row.*

Description

Add an header or footer new row made of one cell. This is a sugar function to be used when you need to add a title row to a flextable, most of the time it will be used in a context of adding a footnote or adding a title on the top line of the flextable.

Usage

```
add_header_lines(x, values = character(0), top = TRUE)
```

```
add_footer_lines(x, values = character(0), top = FALSE)
```

Arguments

x a flextable object

values a character vector, each element will be added a a new row in the header or footer part.

top should the row be inserted at the top or the bottom.

Illustrations**See Also**

Other headers and footers: [add_header_row\(\)](#), [add_header\(\)](#), [set_header_footer_df](#), [set_header_labels\(\)](#)

Examples

```
ft_1 <- flextable( head( iris ) )
ft_1 <- add_header_lines(ft_1, values = "blah blah")
ft_1 <- add_header_lines(ft_1, values = c("blah 1", "blah 2"))
ft_1 <- autofit(ft_1)
ft_1
ft_2 <- flextable( head( iris ) )
ft_2 <- add_footer_lines(ft_2, values = "blah blah")
ft_2 <- add_footer_lines(ft_2, values = c("blah 1", "blah 2"))
ft_2 <- theme_tron(ft_2)
ft_2
```

add_header_row

Add labels and merge cells in a new header or footer row

Description

Add an header or footer new row where some cells are merged, labels are associated with a number of columns to merge. The function is row oriented. One call allow to add one single row.

Usage

```
add_header_row(x, top = TRUE, values = character(0), colwidths = integer(0))
```

```
add_footer_row(x, top = TRUE, values = character(0), colwidths = integer(0))
```

Arguments

x	a flextable object
top	should the row be inserted at the top or the bottom.
values	values to add as a character vector
colwidths	the number of columns to merge in the row for each label

Illustrations

See Also

Other headers and footers: [add_header_lines\(\)](#), [add_header\(\)](#), [set_header_footer_df](#), [set_header_labels\(\)](#)

Examples

```
ft <- flextable( head( iris ) )
ft <- add_header_row(ft, values = "blah blah", colwidths = 5)
ft <- add_header_row(ft, values = c("blah", "blah"), colwidths = c(3,2))
ft <- theme_tron(ft)
ft
ft <- flextable( head( iris ) )
ft <- add_footer_row(ft, values = "blah blah", colwidths = 5)
ft <- add_footer_row(ft, values = c("blah", "blah"), colwidths = c(3,2))
ft
```

align

Set text alignment

Description

change text alignment of selected rows and columns of a flextable.

Usage

```
align(x, i = NULL, j = NULL, align = "left", part = "body")
```

```
align_text_col(x, align = "left", header = TRUE, footer = TRUE)
```

```
align_nottext_col(x, align = "right", header = TRUE, footer = TRUE)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
align	text alignment - a single character value, expected value is one of 'left', 'right', 'center', 'justify'.
part	partname of the table (one of 'all', 'body', 'header', 'footer')
header	should the header be aligned with the body
footer	should the footer be aligned with the body

Illustrations

See Also

Other sugar functions for table style: [bg\(\)](#), [bold\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [font\(\)](#), [highlight\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
ft <- flextable(head(mtcars)[,3:6])
ft <- align(ft, align = "right", part = "all")
ft <- theme_tron_legacy(ft)
ft
ftab <- flextable(mtcars)
ftab <- align_text_col(ftab, align = "left")
ftab <- align_notttext_col(ftab, align = "right")
ftab
```

as_b

bold chunk

Description

The function is producing a chunk with bold font.

Usage

```
as_b(x)
```

Arguments

x value, if a chunk, the chunk will be updated

Illustrations

Note

This is a sugar function that ease the composition of complex labels made of different formatings. It should be used inside a call to [as_paragraph\(\)](#).

See Also

Other chunk elements for paragraph: [as_bracket\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_i\(\)](#), [as_sub\(\)](#), [as_sup\(\)](#), [gg_chunk\(\)](#), [hyperlink_text\(\)](#), [linerange\(\)](#), [lollipop\(\)](#), [minibar\(\)](#), [plot_chunk\(\)](#)

Examples

```
ft <- flextable( head(iris),
  col_keys = c("Sepal.Length", "dummy") )

ft <- compose(ft, j = "dummy",
  value = as_paragraph(
    as_b(Sepal.Length)
  ) )

ft
```

as_bracket

chunk with values in brackets

Description

The function is producing a chunk by pasting values and add the result in brackets. It should be used inside a call to [as_paragraph\(\)](#).

Usage

```
as_bracket(..., sep = ", ", p = "(", s = ")")
```

Arguments

...	text and column names
sep	separator
p	prefix, default to '('
s	suffix, default to ')'

Illustrations**See Also**

Other chunk elements for paragraph: [as_b\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_i\(\)](#), [as_sub\(\)](#), [as_sup\(\)](#), [gg_chunk\(\)](#), [hyperlink_text\(\)](#), [linerange\(\)](#), [lollipop\(\)](#), [minibar\(\)](#), [plot_chunk\(\)](#)

Examples

```
ft <- flextable( head(iris),
  col_keys = c("Species", "Sepal", "Petal") )
ft <- set_header_labels(ft, Sepal="Sepal", Petal="Petal")
ft <- compose(ft, j = "Sepal",
  value = as_paragraph( as_bracket(Sepal.Length, Sepal.Width) ) )
ft <- compose(ft, j = "Petal",
  value = as_paragraph( as_bracket(Petal.Length, Petal.Width) ) )
ft
```

as_chunk	<i>chunk of text wrapper</i>
----------	------------------------------

Description

The function lets add text within flextable objects with function `compose()`. It should be used inside a call to `as_paragraph()`.

Usage

```
as_chunk(x, props = NULL, formatter = format_fun, ...)
```

Arguments

x	text or any element that can be formatted as text with function provided in argument <code>formatter</code> .
props	an <code>officer::fp_text()</code> object to be used to format the text. If not specified, it will be the default value corresponding to the cell.
formatter	a function that will format x as a character vector.
...	additional arguments for <code>formatter</code> function.

Illustrations

See Also

Other chunk elements for paragraph: `as_bracket()`, `as_b()`, `as_image()`, `as_i()`, `as_sub()`, `as_sup()`, `gg_chunk()`, `hyperlink_text()`, `linerange()`, `lollipop()`, `minibar()`, `plot_chunk()`

Examples

```
library(officer)

ft <- flextable( head(iris))

ft <- compose( ft, j = "Sepal.Length",
  value = as_paragraph(
    "Sepal.Length value is ",
    as_chunk(Sepal.Length, props = fp_text(color = "red"))
  ),
  part = "body")
ft <- color(ft, color = "gray40", part = "all")
ft <- autofit(ft)
ft
```

as_flextable	<i>method to convert object to flextable</i>
--------------	--

Description

This is a convenient function to let users create flextable bindings from any objects.

Usage

```
as_flextable(x, ...)
```

Arguments

x	object to be transformed as flextable
...	arguments for custom methods

See Also

Other as_flextable methods: [as_flextable.gam\(\)](#), [as_flextable.glm\(\)](#), [as_flextable.grouped_data\(\)](#), [as_flextable.htest\(\)](#), [as_flextable.lm\(\)](#), [as_flextable.xtable\(\)](#)

as_flextable.gam	<i>tabular summary for gam object</i>
------------------	---------------------------------------

Description

produce a flextable describing a generalized additive model produced by function `mgcv::gam`.

Usage

```
## S3 method for class 'gam'  
as_flextable(x, ...)
```

Arguments

x	gam model
...	unused argument

See Also

Other as_flextable methods: [as_flextable.glm\(\)](#), [as_flextable.grouped_data\(\)](#), [as_flextable.htest\(\)](#), [as_flextable.lm\(\)](#), [as_flextable.xtable\(\)](#), [as_flextable\(\)](#)

Examples

```

if (require("mgcv")) {
  set.seed(2)

  # Simulated data
  dat <- gamSim(1, n = 400, dist = "normal", scale = 2)

  # basic GAM model
  b <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat)

  ft <- as_flextable(b)
  ft
}

```

as_flextable.glm *tabular summary for glm object*

Description

produce a flextable describing a generalized linear model produced by function glm.

Usage

```

## S3 method for class 'glm'
as_flextable(x, ...)

```

Arguments

x	glm model
...	unused argument

Illustrations**See Also**

Other as_flextable methods: [as_flextable.gam\(\)](#), [as_flextable.grouped_data\(\)](#), [as_flextable.htest\(\)](#), [as_flextable.lm\(\)](#), [as_flextable.xtable\(\)](#), [as_flextable\(\)](#)

Examples

```

if(require("broom")){
  dat <- attitude
  dat$high.rating <- (dat$rating > 70)
  probit.model <- glm(high.rating ~ learning + critical +
    advance, data=dat, family = binomial(link = "probit"))
  ft <- as_flextable(probit.model)
  ft
}

```

```
as_flextable.grouped_data
      tabular summary for grouped_data object
```

Description

produce a flextable from a table produced by function `as_grouped_data()`.

Usage

```
## S3 method for class 'grouped_data'
as_flextable(x, col_keys = NULL, hide_grouplabel = FALSE, ...)
```

Arguments

<code>x</code>	object to be transformed as flextable
<code>col_keys</code>	columns names/keys to display. If some column names are not in the dataset, they will be added as blank columns by default.
<code>hide_grouplabel</code>	if TRUE, group label will not be rendered, only level/value will be rendered.
<code>...</code>	unused argument

Illustrations

See Also

[as_grouped_data\(\)](#)

Other `as_flextable` methods: [as_flextable.gam\(\)](#), [as_flextable.glm\(\)](#), [as_flextable.htest\(\)](#), [as_flextable.lm\(\)](#), [as_flextable.xtable\(\)](#), [as_flextable\(\)](#)

Examples

```
library(data.table)
CO2 <- CO2
setDT(CO2)
CO2$conc <- as.integer(CO2$conc)

data_co2 <- dcast(CO2, Treatment + conc ~ Type,
  value.var = "uptake", fun.aggregate = mean)
data_co2 <- as_grouped_data(x = data_co2, groups = c("Treatment"))

ft <- as_flextable( data_co2 )
ft <- add_footer_lines(ft, "dataset CO2 has been used for this flextable")
ft <- add_header_lines(ft, "mean of carbon dioxide uptake in grass plants")
ft <- set_header_labels(ft, conc = "Concentration")
```

```
ft <- autofit(ft)
ft <- width(ft, width = c(1, 1, 1))
ft
```

as_flextable.htest *tabular summary for htest object*

Description

produce a flextable describing an object of class htest.

Usage

```
## S3 method for class 'htest'
as_flextable(x, ...)
```

Arguments

x	htest object
...	unused argument

Illustrations

See Also

Other as_flextable methods: [as_flextable.gam\(\)](#), [as_flextable.glm\(\)](#), [as_flextable.grouped_data\(\)](#), [as_flextable.lm\(\)](#), [as_flextable.xtable\(\)](#), [as_flextable\(\)](#)

Examples

```
if(require("stats")){
  M <- as.table(rbind(c(762, 327, 468), c(484, 239, 477)))
  dimnames(M) <- list(gender = c("F", "M"),
    party = c("Democrat", "Independent", "Republican"))
  ft_1 <- as_flextable(chisq.test(M))
  ft_1
}
```

as_flextable.lm *tabular summary for lm object*

Description

produce a flextable describing a linear model produced by function lm.

Usage

```
## S3 method for class 'lm'  
as_flextable(x, ...)
```

Arguments

x	lm model
...	unused argument

Illustrations

See Also

Other as_flextable methods: [as_flextable.gam\(\)](#), [as_flextable.glm\(\)](#), [as_flextable.grouped_data\(\)](#), [as_flextable.htest\(\)](#), [as_flextable.xtable\(\)](#), [as_flextable\(\)](#)

Examples

```
if(require("broom")){  
  lmod <- lm(rating ~ complaints + privileges +  
            learning + raises + critical, data=attitude)  
  ft <- as_flextable(lmod)  
  ft  
}
```

as_flextable.xtable *get a flextable from a xtable object*

Description

Get a flextable object from a xtable object.

xtable_to_flextable will be deprecated in favor of as_flextable.xtable.

Usage

```
## S3 method for class 'xtable'
as_flextable(
  x,
  text.properties = officer::fp_text(),
  format.args = getOption("xtable.format.args", NULL),
  rowname_col = "rowname",
  hline.after = getOption("xtable.hline.after", c(-1, 0, nrow(x))),
  NA.string = getOption("xtable.NA.string", ""),
  include.rownames = TRUE,
  rotate.colnames = getOption("xtable.rotate.colnames", FALSE),
  ...
)

xtable_to_flextable(
  x,
  text.properties = officer::fp_text(),
  format.args = getOption("xtable.format.args", NULL),
  rowname_col = "rowname",
  hline.after = getOption("xtable.hline.after", c(-1, 0, nrow(x))),
  NA.string = getOption("xtable.NA.string", ""),
  include.rownames = TRUE,
  rotate.colnames = getOption("xtable.rotate.colnames", FALSE),
  ...
)
```

Arguments

x	xtable object
text.properties	default text formatting properties
format.args	List of arguments for the formatC function. See argument format.args of print.xtable. Not yet implemented.
rowname_col	colname used for row names column
hline.after	see ?print.xtable.
NA.string	see ?print.xtable.
include.rownames	see ?print.xtable.
rotate.colnames	see ?print.xtable.
...	unused arguments

Illustrations

See Also

Other `as_flextable` methods: `as_flextable.gam()`, `as_flextable.glm()`, `as_flextable.grouped_data()`, `as_flextable.htest()`, `as_flextable.lm()`, `as_flextable()`

Examples

```
library(officer)
if( require("xtable" )){

  data(tli)
  tli.table <- xtable(tli[1:10, ])
  align(tli.table) <- rep("r", 6)
  align(tli.table) <- "|r|r|c|r|r|"
  ft_1 <- as_flextable(
    tli.table,
    rotate.colnames = TRUE,
    include.rownames = FALSE)
  ft_1 <- height(ft_1, i = 1, part = "header", height = 1)
  ft_1

  Grade3 <- c("A", "B", "B", "A", "B", "C", "C", "D", "A", "B",
    "C", "C", "C", "D", "B", "B", "D", "C", "C", "D")
  Grade6 <- c("A", "A", "A", "B", "B", "B", "B", "B", "C", "C",
    "A", "C", "C", "C", "D", "D", "D", "D", "D", "D")
  Cohort <- table(Grade3, Grade6)
  ft_2 <- as_flextable(xtable(Cohort))
  ft_2 <- set_header_labels(ft_2, rowname = "Grade 3")
  ft_2 <- autofit(ft_2)
  ft_2 <- add_header(ft_2, A = "Grade 6")
  ft_2 <- merge_at(ft_2, i = 1, j = seq_len( ncol(Cohort) ) + 1,
    part = "header" )
  ft_2 <- bold(ft_2, j = 1, bold = TRUE, part = "body")
  ft_2 <- height_all(ft_2, part = "header", height = .4)
  ft_2

  temp.ts <- ts(cumsum(1 + round(rnorm(100), 0)),
    start = c(1954, 7), frequency = 12)
  ft_3 <- as_flextable(x = xtable(temp.ts, digits = 0),
    NA.string = "-")
  ft_3

  detach("package:xtable", unload = TRUE)
}
```

Description

Repeated consecutive values of group columns will be used to define the title of the groups and will be added as a row title.

Usage

```
as_grouped_data(x, groups, columns = NULL)
```

Arguments

x	dataset
groups	columns names to be used as row separators.
columns	columns names to keep

See Also

[as_flextable.grouped_data\(\)](#)

Examples

```
# as_grouped_data -----
library(data.table)
C02 <- C02
setDT(C02)
C02$conc <- as.integer(C02$conc)

data_co2 <- dcast(C02, Treatment + conc ~ Type,
  value.var = "uptake", fun.aggregate = mean)
data_co2
data_co2 <- as_grouped_data(x = data_co2, groups = c("Treatment"))
data_co2
```

as_i

italic chunk

Description

The function is producing a chunk with italic font.

Usage

```
as_i(x)
```

Arguments

x	value, if a chunk, the chunk will be updated
---	--

Illustrations**Note**

This is a sugar function that ease the composition of complex labels made of different formattings. It should be used inside a call to [as_paragraph\(\)](#).

See Also

Other chunk elements for paragraph: [as_bracket\(\)](#), [as_b\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_sub\(\)](#), [as_sup\(\)](#), [gg_chunk\(\)](#), [hyperlink_text\(\)](#), [linerrange\(\)](#), [lollipop\(\)](#), [minibar\(\)](#), [plot_chunk\(\)](#)

Examples

```
ft <- flextable( head(iris),
  col_keys = c("Sepal.Length", "dummy") )

ft <- compose(ft, j = "dummy",
  value = as_paragraph(as_i(Sepal.Length)) )

ft
```

as_image	<i>image chunk wrapper</i>
----------	----------------------------

Description

The function lets add images within flextable objects with function [compose\(\)](#). It should be used inside a call to [as_paragraph\(\)](#).

Usage

```
as_image(src, width = 0.5, height = 0.2, ...)
```

Arguments

src	image filename
width, height	size of the png file in inches
...	unused argument

Illustrations

Note

This chunk option requires package `officedown` in a R Markdown context with Word output format. PowerPoint cannot mix images and text in a paragraph, images are removed when outputting to PowerPoint format.

See Also

`compose()`, `as_paragraph()`

Other chunk elements for paragraph: `as_bracket()`, `as_b()`, `as_chunk()`, `as_i()`, `as_sub()`, `as_sup()`, `gg_chunk()`, `hyperlink_text()`, `linerange()`, `lollipop()`, `minibar()`, `plot_chunk()`

Examples

```
img.file <- file.path( R.home("doc"), "html", "logo.jpg" )
library(officer)

myft <- flextable( head(iris))

myft <- compose( myft, i = 1:3, j = 1,
  value = as_paragraph(
    as_image(src = img.file, width = .20, height = .15),
    " blah blah ",
    as_chunk(Sepal.Length, props = fp_text(color = "red"))
  ),
  part = "body")

ft <- autofit(myft)
ft
```

as_paragraph

concatenate chunks in a flextable

Description

The function is concatenating text and images within paragraphs of a flextable object, this function is to be used with function `compose()`.

Usage

```
as_paragraph(..., list_values = NULL)
```

Arguments

`...` chunk elements that are defining paragraph

`list_values` a list of chunk elements that are defining paragraph. If specified argument `...` is unused.

Illustrations**See Also**

[as_chunk\(\)](#), [minibar\(\)](#), [as_image\(\)](#), [hyperlink_text\(\)](#)

Examples

```
library(officer)
ft <- flextable( head(iris, n = 10 ))

ft <- compose(ft, j = 1,
  value = as_paragraph(
    minibar(value = Sepal.Length, max = max(Sepal.Length)),
    " ",
    as_chunk( Sepal.Length, formatter = formatC,
      props = fp_text(color = "orange") ),
    " blah blah"
  ),
  part = "body")

ft <- autofit(ft)
ft
```

as_raster

get a flextable as a raster

Description

save a flextable as an image and return the corresponding raster. This function has been implemented to let flextable be printed on a ggplot object.

Usage

```
as_raster(x, zoom = 2, expand = 2, webshot = "webshot")
```

Arguments

x	a flextable object
zoom, expand	parameters used by webshot function.
webshot	webshot package as a scalar character, one of "webshot" or "webshot2".

Note

This function requires packages: webshot and magick.

See Also

Other flextable print function: [flextable_to_rmd\(\)](#), [htmltools_value\(\)](#), [knit_print.flextable\(\)](#), [plot.flextable\(\)](#), [print.flextable\(\)](#), [save_as_docx\(\)](#), [save_as_html\(\)](#), [save_as_image\(\)](#), [save_as_pptx\(\)](#)

Examples

```
ft <- qflextable( head( mtcars ) )
## Not run:
if( require("ggplot2") && require("webshot") ){
  print(qplot(speed, dist, data = cars, geom = "point"))
  grid::grid.raster(as_raster(ft))
}

## End(Not run)
```

as_sub

subscript chunk

Description

The function is producing a chunk with subscript vertical alignment.

Usage

```
as_sub(x)
```

Arguments

x value, if a chunk, the chunk will be updated

Illustrations**Note**

This is a sugar function that ease the composition of complex labels made of different formatings. It should be used inside a call to [as_paragraph\(\)](#).

See Also

Other chunk elements for paragraph: [as_bracket\(\)](#), [as_b\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_i\(\)](#), [as_sup\(\)](#), [gg_chunk\(\)](#), [hyperlink_text\(\)](#), [linerange\(\)](#), [lollipop\(\)](#), [minibar\(\)](#), [plot_chunk\(\)](#)

Examples

```
ft <- flextable( head(iris), col_keys = c("dummy") )

ft <- compose(ft, i = 1, j = "dummy", part = "header",
  value = as_paragraph(
    as_sub("Sepal.Length"),
    " anything "
  ) )

ft <- autofit(ft)
ft
```

as_sup

*superscript chunk***Description**

The function is producing a chunk with superscript vertical alignment.

Usage

```
as_sup(x)
```

Arguments

x value, if a chunk, the chunk will be updated

Illustrations**Note**

This is a sugar function that ease the composition of complex labels made of different formattings. It should be used inside a call to [as_paragraph\(\)](#).

See Also

Other chunk elements for paragraph: [as_bracket\(\)](#), [as_b\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_i\(\)](#), [as_sub\(\)](#), [gg_chunk\(\)](#), [hyperlink_text\(\)](#), [linerange\(\)](#), [lollipop\(\)](#), [minibar\(\)](#), [plot_chunk\(\)](#)

Examples

```
ft <- flextable( head(iris), col_keys = c("dummy") )

ft <- compose(ft, i = 1, j = "dummy", part = "header",
  value = as_paragraph(
    " anything ",
    as_sup("Sepal.Width")
  ) )
```

```

    ) )

ft <- autofit(ft)
ft

```

autofit

Adjusts cell widths and heights

Description

compute and apply optimized widths and heights (minimum estimated widths and heights for each table columns and rows in inches returned by function [dim_pretty\(\)](#)).

This function is to be used when the table widths and heights should automatically be adjusted to fit the size of the content.

Usage

```
autofit(x, add_w = 0.1, add_h = 0.1, part = c("body", "header"))
```

Arguments

x	flextable object
add_w	extra width to add in inches
add_h	extra height to add in inches
part	partname of the table (one of 'all', 'body', 'header' or 'footer')

line breaks

Soft returns (a line break in a paragraph) are not supported. Function autofit will return wrong results if `\n` are used (they will be considered as "").

Illustrations

Note

This function is not related to 'Microsoft Word' *Autofit* feature.

See Also

Other flextable dimensions: [dim.flextable\(\)](#), [dim_pretty\(\)](#), [fit_to_width\(\)](#), [flextable_dim\(\)](#), [height\(\)](#), [hrule\(\)](#), [set_table_properties\(\)](#), [width\(\)](#)

Examples

```
ft_1 <- flextable(head(mtcars))
ft_1
ft_2 <- autofit(ft_1)
ft_2
```

bg *Set background color*

Description

change background color of selected rows and columns of a flextable.

Usage

```
bg(x, i = NULL, j = NULL, bg, part = "body", source = j)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
bg	color to use as background color. If a function, function need to return a character vector of colors.
part	partname of the table (one of 'all', 'body', 'header', 'footer')
source	if bg is a function, source is specifying the dataset column to be used as argument to bg. This is only useful if j is colored with values contained in another (or other) column.

Illustrations**See Also**

Other sugar functions for table style: [align\(\)](#), [bold\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [font\(\)](#), [highlight\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
ft_1 <- flextable(head(mtcars))
ft_1 <- bg(ft_1, bg = "wheat", part = "header")
ft_1 <- bg(ft_1, i = ~ qsec < 18, bg = "#EFEFEF", part = "body")
ft_1 <- bg(ft_1, j = "drat", bg = "#606060", part = "all")
ft_1 <- color(ft_1, j = "drat", color = "white", part = "all")
ft_1
```

```

if(require("scales")){
  ft_2 <- flextable(head(iris))
  colourer <- col_numeric(
    palette = c("wheat", "red"),
    domain = c(0, 7))
  ft_2 <- bg(ft_2, j = c("Sepal.Length", "Sepal.Width",
    "Petal.Length", "Petal.Width"),
    bg = colourer, part = "body")
  ft_2
}

```

body_add_flextable *add flextable into a Word document*

Description

add a flextable into a Word document.

Usage

```
body_add_flextable(x, value, align = "center", pos = "after", split = FALSE)
```

```

body_replace_flextable_at_bkm(
  x,
  bookmark,
  value,
  align = "center",
  split = FALSE
)

```

Arguments

x	an rdocx object
value	flextable object
align	left, center (default) or right.
pos	where to add the flextable relative to the cursor, one of "after", "before", "on" (end of line).
split	set to TRUE if you want to activate Word option 'Allow row to break across pages'.
bookmark	bookmark id

body_replace_flextable_at_bkm

Use this function if you want to replace a paragraph containing a bookmark with a flextable. As a side effect, the bookmark will be lost.

Examples

```

library(officer)

# autonum for caption
autonum <- run_autonum(seq_id = "tab", bkm = "mtcars")

ftab <- flextable( head( mtcars ) )
ftab <- set_caption(ftab, caption = "mtcars data", autonum = autonum)
ftab <- autofit(ftab)
doc <- read_docx()
doc <- body_add_flextable(doc, value = ftab)
fileout <- tempfile(fileext = ".docx")
# fileout <- "test.docx" # uncomment to write in your working directory
print(doc, target = fileout)

```

bold*Set bold font*

Description

change font weight of selected rows and columns of a flextable.

Usage

```
bold(x, i = NULL, j = NULL, bold = TRUE, part = "body")
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
bold	boolean value
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations**See Also**

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [font\(\)](#), [highlight\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```

ft <- flextable(head(iris))
ft <- bold(ft, bold = TRUE, part = "header")

```

border_inner	<i>set vertical & horizontal inner borders</i>
--------------	--

Description

The function is applying a vertical and horizontal borders to inner content of one or all parts of a flextable.

Usage

```
border_inner(x, border = NULL, part = "all")
```

Arguments

x	a flextable object
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner_v\(\)](#), [border_outer\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline_top\(\)](#), [hline\(\)](#), [vline_left\(\)](#), [vline_right\(\)](#), [vline\(\)](#)

Examples

```
library(officer)
std_border = fp_border(color="orange", width = 1)

dat <- iris[c(1:5, 51:55, 101:105),]
ft <- flextable(dat)
ft <- border_remove(x = ft)

# add inner vertical borders
ft <- border_inner(ft, border = std_border )
ft
```

border_inner_h	<i>set inner borders</i>
----------------	--------------------------

Description

The function is applying a border to inner content of one or all parts of a flextable.

Usage

```
border_inner_h(x, border = NULL, part = "body")
```

Arguments

x	a flextable object
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_v\(\)](#), [border_inner\(\)](#), [border_outer\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline_top\(\)](#), [hline\(\)](#), [vline_left\(\)](#), [vline_right\(\)](#), [vline\(\)](#)

Examples

```
library(officer)
std_border = fp_border(color="orange", width = 1)

dat <- iris[c(1:5, 51:55, 101:105),]
ft <- flextable(dat)
ft <- border_remove(x = ft)

# add inner horizontal borders
ft <- border_inner_h(ft, border = std_border )
ft
```

border_inner_v	<i>set vertical inner borders</i>
----------------	-----------------------------------

Description

The function is applying a vertical border to inner content of one or all parts of a flextable.

Usage

```
border_inner_v(x, border = NULL, part = "all")
```

Arguments

x	a flextable object
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner\(\)](#), [border_outer\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline_top\(\)](#), [hline\(\)](#), [vline_left\(\)](#), [vline_right\(\)](#), [vline\(\)](#)

Examples

```
library(officer)
std_border = fp_border(color="orange", width = 1)

dat <- iris[c(1:5, 51:55, 101:105),]
ft <- flextable(dat)
ft <- border_remove(x = ft)

# add inner vertical borders
ft <- border_inner_v(ft, border = std_border )
ft
```

border_outer	<i>set outer borders</i>
--------------	--------------------------

Description

The function is applying a border to outer cells of one or all parts of a flextable.

Usage

```
border_outer(x, border = NULL, part = "all")
```

Arguments

x	a flextable object
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner_v\(\)](#), [border_inner\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline_top\(\)](#), [hline\(\)](#), [vline_left\(\)](#), [vline_right\(\)](#), [vline\(\)](#)

Examples

```
library(officer)
big_border = fp_border(color="red", width = 2)

dat <- iris[c(1:5, 51:55, 101:105),]
ft <- flextable(dat)
ft <- border_remove(x = ft)

# add outer borders
ft <- border_outer(ft, part="all", border = big_border )
ft
```

border_remove	<i>remove borders</i>
---------------	-----------------------

Description

The function is deleting all borders of the flextable object.

Usage

```
border_remove(x)
```

Arguments

x a flextable object

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner_v\(\)](#), [border_inner\(\)](#), [border_outer\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline_top\(\)](#), [hline\(\)](#), [vline_left\(\)](#), [vline_right\(\)](#), [vline\(\)](#)

Examples

```
dat <- iris[c(1:5, 51:55, 101:105),]
ft_1 <- flextable(dat)
ft_1 <- theme_box(ft_1)
ft_1

# remove all borders
ft_2 <- border_remove(x = ft_1)
ft_2
```

colformat_char	<i>format character cells</i>
----------------	-------------------------------

Description

Format character cells in a flextable.

Usage

```
colformat_char(
  x,
  i = NULL,
  j = NULL,
  na_str = get_flextable_defaults()$na_str,
  prefix = "",
  suffix = ""
)
```

Arguments

x a flextable object
i rows selection
j columns selection.
na_str string to be used for NA values
prefix, suffix string to be used as prefix or suffix

See Also

Other cells formatters: [colformat_datetime\(\)](#), [colformat_date\(\)](#), [colformat_double\(\)](#), [colformat_image\(\)](#), [colformat_int\(\)](#), [colformat_lgl\(\)](#), [colformat_num\(\)](#), [compose\(\)](#), [set_formatter\(\)](#)

Examples

```
dat <- iris
z <- flextable(head(dat))
ft <- colformat_char(
  x = z, j = "Species", suffix = "!")
z <- autofit(z)
z
```

colformat_date *format date cells*

Description

Format date cells in a flextable.

Usage

```
colformat_date(
  x,
  i = NULL,
  j = NULL,
  fmt_date = get_flextable_defaults()$fmt_date,
```

```

na_str = get_flextable_defaults()$na_str,
prefix = "",
suffix = ""
)

```

Arguments

x	a flextable object
i	rows selection
j	columns selection.
fmt_date	see strptime()
na_str	string to be used for NA values
prefix	string to be used as prefix or suffix
suffix	string to be used as prefix or suffix

See Also

Other cells formatters: [colformat_char\(\)](#), [colformat_datetime\(\)](#), [colformat_double\(\)](#), [colformat_image\(\)](#), [colformat_int\(\)](#), [colformat_lgl\(\)](#), [colformat_num\(\)](#), [compose\(\)](#), [set_formatter\(\)](#)

Examples

```

dat <- data.frame(z = Sys.Date() + 1:3,
  w = Sys.Date() - 1:3)
ft <- flextable(dat)
ft <- colformat_date(x = ft)
ft <- autofit(ft)
ft

```

colformat_datetime *format datetime cells*

Description

Format datetime cells in a flextable.

Usage

```

colformat_datetime(
  x,
  i = NULL,
  j = NULL,
  fmt_datetime = get_flextable_defaults()$fmt_datetime,
  na_str = get_flextable_defaults()$na_str,
  prefix = "",
  suffix = ""
)

```

Arguments

x	a flextable object
i	rows selection
j	columns selection.
fmt_datetime	see strptime()
na_str	string to be used for NA values
prefix	string to be used as prefix or suffix
suffix	string to be used as prefix or suffix

See Also

Other cells formatters: [colformat_char\(\)](#), [colformat_date\(\)](#), [colformat_double\(\)](#), [colformat_image\(\)](#), [colformat_int\(\)](#), [colformat_lgl\(\)](#), [colformat_num\(\)](#), [compose\(\)](#), [set_formatter\(\)](#)

Examples

```
dat <- data.frame(z = Sys.time() + (1:3)*24,  
  w = Sys.Date() - (1:3)*24)  
ft <- flextable(dat)  
ft <- colformat_datetime(x = ft)  
ft <- autofit(ft)  
ft
```

colformat_double	<i>format numeric cells</i>
------------------	-----------------------------

Description

Format numeric cells in a flextable.

Usage

```
colformat_double(  
  x,  
  i = NULL,  
  j = NULL,  
  big.mark = get_flextable_defaults()$big.mark,  
  decimal.mark = get_flextable_defaults()$decimal.mark,  
  digits = get_flextable_defaults()$digits,  
  na_str = get_flextable_defaults()$na_str,  
  prefix = "",  
  suffix = ""  
)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection.
big.mark, digits, decimal.mark	see format()
na_str	string to be used for NA values
prefix	string to be used as prefix or suffix
suffix	string to be used as prefix or suffix

See Also

Other cells formatters: [colformat_char\(\)](#), [colformat_datetime\(\)](#), [colformat_date\(\)](#), [colformat_image\(\)](#), [colformat_int\(\)](#), [colformat_lgl\(\)](#), [colformat_num\(\)](#), [compose\(\)](#), [set_formatter\(\)](#)

Examples

```
dat <- mtcars
ft <- flextable(head(dat))
ft <- colformat_double(x = ft,
  big.mark=",", digits = 2, na_str = "N/A")
autofit(ft)
```

colformat_image	<i>format cells as images</i>
-----------------	-------------------------------

Description

Format image paths as images in a flextable.

Usage

```
colformat_image(
  x,
  i = NULL,
  j = NULL,
  width,
  height,
  na_str = get_flextable_defaults()$na_str,
  prefix = "",
  suffix = ""
)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection.
width, height	size of the png file in inches
na_str	string to be used for NA values
prefix	string to be used as prefix or suffix
suffix	string to be used as prefix or suffix

See Also

Other cells formatters: [colformat_char\(\)](#), [colformat_datetime\(\)](#), [colformat_date\(\)](#), [colformat_double\(\)](#), [colformat_int\(\)](#), [colformat_lgl\(\)](#), [colformat_num\(\)](#), [compose\(\)](#), [set_formatter\(\)](#)

Examples

```
img.file <- file.path( R.home("doc"), "html", "logo.jpg" )

dat <- head(iris)
dat$Species <- as.character(dat$Species)
dat[c(1, 3, 5), "Species"] <- img.file

myft <- flextable( dat)
myft <- colformat_image(
  myft, i = c(1, 3, 5),
  j = "Species", width = .20, height = .15)
ft <- autofit(myft)
ft
```

colformat_int *format integer cells*

Description

Format integer cells in a flextable.

Usage

```
colformat_int(
  x,
  i = NULL,
  j = NULL,
  big.mark = get_flextable_defaults()$big.mark,
  na_str = get_flextable_defaults()$na_str,
  prefix = "",
  suffix = ""
)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection.
big.mark	see <code>format()</code>
na_str	string to be used for NA values
prefix	string to be used as prefix or suffix
suffix	string to be used as prefix or suffix

See Also

Other cells formatters: `colformat_char()`, `colformat_datetime()`, `colformat_date()`, `colformat_double()`, `colformat_image()`, `colformat_lgl()`, `colformat_num()`, `compose()`, `set_formatter()`

Examples

```
z <- flextable(head(mtcars))
j <- c("vs", "am", "gear", "carb")
z <- colformat_int(x = z, j = j, prefix = "# ")
z
```

`colformat_lgl`*format logical cells*

Description

Format logical cells in a flextable.

Usage

```
colformat_lgl(  
  x,  
  i = NULL,  
  j = NULL,  
  true = "true",  
  false = "false",  
  na_str = get_flextable_defaults()$na_str,  
  prefix = "",  
  suffix = ""  
)
```


Arguments

x	a flextable object
i	rows selection
j	columns selection.
false, true	string to be used for logical
na_str	string to be used for NA values
prefix	string to be used as prefix or suffix
suffix	string to be used as prefix or suffix

See Also

Other cells formatters: [colformat_char\(\)](#), [colformat_datetime\(\)](#), [colformat_date\(\)](#), [colformat_double\(\)](#), [colformat_image\(\)](#), [colformat_int\(\)](#), [colformat_num\(\)](#), [compose\(\)](#), [set_formatter\(\)](#)

Examples

```
dat <- data.frame(a = c(TRUE, FALSE), b = c(FALSE, TRUE))

z <- flextable(dat)
z <- colformat_lgl(x = z, j = c("a", "b"))
autofit(z)
```

colformat_num	<i>format numeric cells</i>
---------------	-----------------------------

Description

Format numeric cells in a flextable.

Usage

```
colformat_num(  
  x,  
  i = NULL,  
  j = NULL,  
  big.mark = get_flextable_defaults()$big.mark,  
  decimal.mark = get_flextable_defaults()$decimal.mark,  
  digits = get_flextable_defaults()$digits,  
  na_str = get_flextable_defaults()$na_str,  
  prefix = "",  
  suffix = ""  
)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection.
big.mark, digits, decimal.mark	see format()
na_str	string to be used for NA values
prefix	string to be used as prefix or suffix
suffix	string to be used as prefix or suffix

Illustrations

See Also

Other cells formatters: [colformat_char\(\)](#), [colformat_datetime\(\)](#), [colformat_date\(\)](#), [colformat_double\(\)](#), [colformat_image\(\)](#), [colformat_int\(\)](#), [colformat_lgl\(\)](#), [compose\(\)](#), [set_formatter\(\)](#)

Examples

```
dat <- mtcars
ft <- flextable(head(dat))
ft <- colformat_num(x = ft,
  big.mark="," , digits = 2, na_str = "N/A")
ft <- autofit(ft)
ft
```

color	<i>Set font color</i>
-------	-----------------------

Description

change font color of selected rows and columns of a flextable.

Usage

```
color(x, i = NULL, j = NULL, color, part = "body", source = j)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
color	color to use as font color. If a function, function need to return a character vector of colors.
part	partname of the table (one of 'all', 'body', 'header', 'footer')
source	if bg is a function, source is specifying the dataset column to be used as argument to color. This is only useful if j is colored with values contained in another (or other) column.

Illustrations**See Also**

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [bold\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [font\(\)](#), [highlight\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
ft <- flextable(head(mtcars))
ft <- color(ft, color = "orange", part = "header")
ft <- color(ft, color = "red",
  i = ~ qsec < 18 & vs < 1 )
ft

if(require("scales")){
scale <- scales::col_numeric(domain= c(-1, 1), palette = "RdBu")
x <- as.data.frame(cor(iris[-5]))
x <- cbind(
  data.frame(colname = colnames(x),
    stringsAsFactors = FALSE),
  x)

ft_2 <- flextable(x)
ft_2 <- color(ft_2, j = x$colname, color = scale)
ft_2 <- set_formatter_type(ft_2)
ft_2
}
```

`compose`*Define flextable displayed values*

Description

Modify flextable displayed values. Function is handling complex formatting as well as image insertion.

Function `mk_par` is another name for `compose` as there is an unwanted conflict with package `purrr`.

Usage

```
compose(x, i = NULL, j = NULL, value, part = "body")
```

```
mk_par(x, i = NULL, j = NULL, value, part = "body")
```

Arguments

<code>x</code>	a flextable object
<code>i</code>	rows selection
<code>j</code>	column selection
<code>value</code>	a call to function as_paragraph() .
<code>part</code>	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other cells formatters: [colformat_char\(\)](#), [colformat_datetime\(\)](#), [colformat_date\(\)](#), [colformat_double\(\)](#), [colformat_image\(\)](#), [colformat_int\(\)](#), [colformat_lgl\(\)](#), [colformat_num\(\)](#), [set_formatter\(\)](#)

Examples

```
library(officer)
ft <- flextable(head( mtcars, n = 10))
ft <- compose(ft, j = "carb", i = ~ drat > 3.5,
  value = as_paragraph("carb is ", as_chunk( sprintf("%.1f", carb)) )
)
ft <- autofit(ft)
```

continuous_summary	<i>continuous columns summary</i>
--------------------	-----------------------------------

Description

create a data.frame summary for continuous variables

Usage

```
continuous_summary(  
  dat,  
  columns = NULL,  
  by = character(0),  
  hide_grouplabel = TRUE,  
  digits = 3  
)
```

Arguments

<code>dat</code>	a data.frame
<code>columns</code>	continuous variables to be summarized. If NULL all continuous variables are summarized.
<code>by</code>	discrete variables to use as groups when summarizing.
<code>hide_grouplabel</code>	if TRUE, group label will not be rendered, only level/value will be rendered.
<code>digits</code>	the desired number of digits after the decimal point

Illustrations

Examples

```
ft_1 <- continuous_summary(iris, names(iris)[1:4], by = "Species",  
  hide_grouplabel = FALSE)  
ft_1
```

delete_part	<i>delete flextable part</i>
-------------	------------------------------

Description

indicate to not print a part of the flextable, i.e. an header, footer or the body.

Usage

```
delete_part(x, part = "header")
```

Arguments

x	a flextable object
part	partname of the table to delete (one of 'body', 'header' or 'footer').

Illustrations**Examples**

```
ft <- flextable( head( iris ) )
ft <- delete_part(x = ft, part = "header")
ft
```

dim.flextable	<i>Get widths and heights of flextable</i>
---------------	--

Description

returns widths and heights for each table columns and rows. Values are expressed in inches.

Usage

```
## S3 method for class 'flextable'
dim(x)
```

Arguments

x	flextable object
---	------------------

See Also

Other flextable dimensions: [autofit\(\)](#), [dim_pretty\(\)](#), [fit_to_width\(\)](#), [flextable_dim\(\)](#), [height\(\)](#), [hrule\(\)](#), [set_table_properties\(\)](#), [width\(\)](#)

Examples

```
ftab <- flextable(head(iris))
dim(ftab)
```

dim_pretty

Calculate pretty dimensions

Description

return minimum estimated widths and heights for each table columns and rows in inches.

Usage

```
dim_pretty(x, part = "all")
```

Arguments

x	flextable object
part	partname of the table (one of 'all', 'body', 'header' or 'footer')

line breaks

Soft returns (a line break in a paragraph) are not supported. Function dim_pretty will return wrong results if \n are used (they will be considered as "").

See Also

Other flextable dimensions: [autofit\(\)](#), [dim.flextable\(\)](#), [fit_to_width\(\)](#), [flextable_dim\(\)](#), [height\(\)](#), [hrule\(\)](#), [set_table_properties\(\)](#), [width\(\)](#)

Examples

```
ftab <- flextable(head(mtcars))
dim_pretty(ftab)
```

`empty_blanks`*make blank columns as transparent*

Description

blank columns are set as transparent. This is a shortcut function that will delete top and bottom borders, change background color to transparent and display empty content.

Usage

```
empty_blanks(x)
```

Arguments

`x` a flextable object

See Also

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [bold\(\)](#), [color\(\)](#), [fontsize\(\)](#), [font\(\)](#), [highlight\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
typology <- data.frame(
  col_keys = c( "Sepal.Length", "Sepal.Width", "Petal.Length",
               "Petal.Width", "Species" ),
  what = c("Sepal", "Sepal", "Petal", "Petal", " " ),
  measure = c("Length", "Width", "Length", "Width", "Species"),
  stringsAsFactors = FALSE )
typology

ftab <- flextable(head(iris), col_keys = c("Species",
    "break1", "Sepal.Length", "Sepal.Width",
    "break2", "Petal.Length", "Petal.Width" ) )
ftab <- set_header_df(ftab, mapping = typology, key = "col_keys" )
ftab <- merge_h(ftab, part = "header")
ftab <- theme_vanilla(ftab)
ftab <- empty_blanks(ftab)
ftab <- width(ftab, j = c(2, 5), width = .1 )
ftab
```

fit_to_width	<i>fit a flextable to a maximum width</i>
--------------	---

Description

decrease font size for each cell incrementally until it fits a given max_width.

Usage

```
fit_to_width(x, max_width, inc = 1L, max_iter = 20)
```

Arguments

x	flextable object
max_width	maximum width to fit in inches
inc	the font size decrease for each step
max_iter	maximum iterations

Illustrations

See Also

Other flextable dimensions: [autofit\(\)](#), [dim.flextable\(\)](#), [dim_pretty\(\)](#), [flextable_dim\(\)](#), [height\(\)](#), [hrule\(\)](#), [set_table_properties\(\)](#), [width\(\)](#)

Examples

```
ft_1 <- qflextable(head(mtcars))
ft_1 <- padding(ft_1, padding = 0, part = "all")
ft_1 <- width(ft_1, width = 1)
ft_1

ft_2 <- fit_to_width(ft_1, max_width = 5.5)
ft_2
```

fix_border_issues	<i>fix border issues when cell are merged</i>
-------------------	---

Description

When cells are merged, the rendered borders will be those of the first cell. If a column is made of three merged cells, the bottom border that will be seen will be the bottom border of the first cell in the column. From a user point of view, this is wrong, the bottom should be the one defined for cell 3. This function modify the border values to avoid that effect.

Usage

```
fix_border_issues(x, part = "all")
```

Arguments

x	flextable object
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Examples

```
library(officer)
dat <- data.frame(a = 1:5, b = 6:10)
ft <- flextable(dat)
ft <- theme_box(ft)
ft <- merge_at(ft, i = 4:5, j = 1, part = "body")
ft <- hline(ft, i = 5, part = "body",
  border = fp_border(color = "red", width = 5) )
print(ft)
ft <- fix_border_issues(ft)
print(ft)
```

flextable	<i>flextable creation</i>
-----------	---------------------------

Description

Create a flextable object with function `flextable`.

`flextable` are designed to make tabular reporting easier for R users. Functions are available to let you format text, paragraphs and cells; table cells can be merge vertically or horizontally, row headers can easilly be defined, rows heights and columns widths can be manually set or automatically computed.

Default formatting properties are automatically applied to every flextable you produce. You can change these default values with function `set_flextable_defaults()`.

Usage

```
flextable(  
  data,  
  col_keys = names(data),  
  cwidth = 0.75,  
  cheight = 0.25,  
  defaults = list(),  
  theme_fun = theme_booktabs  
)
```

```
qflexible(data)
```

```
regulartable(data, col_keys = names(data), cwidth = 0.75, cheight = 0.25)
```

Arguments

data	dataset
col_keys	columns names/keys to display. If some column names are not in the dataset, they will be added as blank columns by default.
cwidth, cheight	initial width and height to use for cell sizes in inches.
defaults, theme_fun	deprecated, use set_flexible_defaults() instead.

Details

A flextable is made of 3 parts: header, body and footer.

Most functions have an argument named `part` that will be used to specify what part of the table should be modified.

Illustrations**qflexible**

`qflexible` is a convenient tool to produce quickly a flextable for reporting where layout is fixed and columns widths adjusted with [autofit\(\)](#).

Note

Function `regulartable` is maintained for compatibility with old codes made by users but be aware it produces the same exact object than `flextable`. This function should be deprecated then removed in the next versions.

See Also

[style\(\)](#), [autofit\(\)](#), [theme_booktabs\(\)](#), [knit_print.flexible\(\)](#), [compose\(\)](#), [footnote\(\)](#), [set_caption\(\)](#)

Examples

```
ft <- flextable(head(mtcars))
ft
```

flextable_dim	<i>width and height of a flextable object</i>
---------------	---

Description

Returns the width, height and aspect ratio of a flextable in a named list. The width and height are in inches. The aspect ratio is the ratio corresponding to height/width.

Names of the list are width, height and aspect_ratio.

Usage

```
flextable_dim(x)
```

Arguments

x a flextable object

See Also

Other flextable dimensions: [autofit\(\)](#), [dim.flextable\(\)](#), [dim_pretty\(\)](#), [fit_to_width\(\)](#), [height\(\)](#), [hrule\(\)](#), [set_table_properties\(\)](#), [width\(\)](#)

Examples

```
ftab <- flextable(head(iris))
flextable_dim(ftab)
ftab <- autofit(ftab)
flextable_dim(ftab)
```

flextable_html_dependency	<i>htmlDependency for flextable objects</i>
---------------------------	---

Description

When using loops in an R Markdown for HTML document, the htmlDependency object for flextable must also be added at least once.

Usage

```
flextable_html_dependency()
```

Examples

```
if(require("htmltools"))
  div(flextable_html_dependency())
```

flextable_to_rmd *flextable raw code*

Description

Print openxml, latex or html code of a flextable. The function is particularly useful when you want to generate flextable in a loop from a R Markdown document.

Inside R Markdown document, chunk option results must be set to 'asis'.

All arguments whose name starts with `ft.` can be set in the chunk options.

See [knit_print.flextable](#) for more details.

Usage

```
flextable_to_rmd(
  x,
  ft.align = opts_current$get("ft.align"),
  ft.split = opts_current$get("ft.split"),
  ft.tabcolsep = opts_current$get("ft.tabcolsep"),
  ft.arraystretch = opts_current$get("ft.arraystretch"),
  ft.left = opts_current$get("ft.left"),
  ft.top = opts_current$get("ft.top"),
  webshot = opts_current$get("webshot"),
  bookdown = FALSE,
  pandoc2 = TRUE,
  print = TRUE
)
```

Arguments

<code>x</code>	a flextable object
<code>ft.align</code>	flextable alignment, supported values are 'left', 'center' and 'right'.
<code>ft.split</code>	Word option 'Allow row to break across pages' can be activated when TRUE.
<code>ft.tabcolsep</code>	space between the text and the left/right border of its containing cell, the default value is 8 points.
<code>ft.arraystretch</code>	height of each row relative to its default height, the default value is 1.5.
<code>ft.left</code> , <code>ft.top</code>	Position should be defined with options <code>ft.left</code> and <code>ft.top</code> . These are the top left coordinates in inches of the placeholder that will contain the table. Their default values are 1 and 2 inches.

webshot	webshot package as a scalar character, one of "webshot" or "webshot2".
bookdown	TRUE or FALSE (default) to support cross referencing with bookdown.
pandoc2	TRUE (default) or FALSE to get the string in a pandoc raw HTML attribute (only valid when pandoc version is ≥ 2).
print	print output if TRUE

See Also

Other flextable print function: [as_raster\(\)](#), [htmltools_value\(\)](#), [knit_print.flextable\(\)](#), [plot.flextable\(\)](#), [print.flextable\(\)](#), [save_as_docx\(\)](#), [save_as_html\(\)](#), [save_as_image\(\)](#), [save_as_pptx\(\)](#)

Examples

```
demo_loop <- system.file(package = "flextable", "examples/rmd", "loop_with_flextable.Rmd")
rmd_file <- tempfile(fileext = ".Rmd")
file.copy(demo_loop, to = rmd_file, overwrite = TRUE)
rmd_file # R Markdown document used for demo
if(require("rmarkdown", quietly = TRUE)){
# render(input = rmd_file, output_format = "word_document",
#   output_file = "loop_with_flextable.docx")
# render(input = rmd_file, output_format = "html_document",
#   output_file = "loop_with_flextable.html")
# render(input = rmd_file,
#   output_format = rmarkdown::pdf_document(latex_engine = "xelatex"),
#   output_file = "loop_with_flextable.pdf")
}
```

font	<i>Set font</i>
------	-----------------

Description

change font of selected rows and columns of a flextable.

Usage

```
font(x, i = NULL, j = NULL, fontname, part = "body")
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
fontname	string value, the font name.
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations**See Also**

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [bold\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [highlight\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
require("gdttools")
fontname <- "Brush Script MT"

if( font_family_exists(fontname) ){
  ft_1 <- flextable(head(iris))
  ft_2 <- font(ft_1, fontname = fontname, part = "header")
  ft_2 <- font(ft_2, fontname = fontname, j = 5)
  ft_2
}
```

fontsize

Set font size

Description

change font size of selected rows and columns of a flextable.

Usage

```
fontsize(x, i = NULL, j = NULL, size = 11, part = "body")
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
size	integer value (points)
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations**See Also**

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [bold\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [font\(\)](#), [highlight\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
ft <- flextable(head(iris))
ft <- fontsize(ft, size = 14, part = "header")
ft <- fontsize(ft, size = 14, j = 2)
ft <- fontsize(ft, size = 7, j = 3)
ft
```

footers_flextable_at_bkm

add flextable at a bookmark location in document's footer

Description

replace in the footer of a document a paragraph containing a bookmark by a flextable. A bookmark will be considered as valid if enclosing words within a paragraph; i.e., a bookmark along two or more paragraphs is invalid, a bookmark set on a whole paragraph is also invalid, but bookmarking few words inside a paragraph is valid.

Usage

```
footers_flextable_at_bkm(x, bookmark, value)
```

Arguments

x	an rdocx object
bookmark	bookmark id
value	a flextable object

footnote

add footnotes to flextable

Description

add footnotes to a flextable object. A symbol is appended where the footnote is defined and the note is appended in the footer part of the table.

Usage

```
footnote(
  x,
  i = NULL,
  j = NULL,
  value,
  ref_symbols = NULL,
  part = "body",
  inline = FALSE,
  sep = "; "
)
```


Arguments

x	a flextable object
i	rows selection
j	column selection
value	a call to function <code>as_paragraph()</code> .
ref_symbols	character value, symbols to append that will be used as references to notes.
part	partname of the table (one of 'body', 'header', 'footer')
inline	whether to add footnote on same line as previous footnote or not
sep	inline = T, character string to use as a separator between footnotes

Illustrations**Examples**

```
ft_1 <- flextable(head(iris))
ft_1 <- footnote( ft_1, i = 1, j = 1:3,
  value = as_paragraph(
    c("This is footnote one",
      "This is footnote two",
      "This is footnote three")
  ),
  ref_symbols = c("a", "b", "c"),
  part = "header")
ft_1 <- valign(ft_1, valign = "bottom", part = "header")
ft_1 <- autofit(ft_1)

ft_2 <- flextable(head(iris))
ft_2 <- autofit(ft_2)
ft_2 <- footnote( ft_2, i = 1, j = 1:2,
  value = as_paragraph(
    c("This is footnote one",
      "This is footnote two")
  ),
  ref_symbols = c("a", "b"),
  part = "header", inline = TRUE)
ft_2 <- footnote( ft_2, i = 1, j = 3:4,
  value = as_paragraph(
    c("This is footnote three",
      "This is footnote four")
  ),
  ref_symbols = c("c", "d"),
  part = "header", inline = TRUE)

ft_2
```

fp_border_default *Border formatting properties*

Description

Create a `fp_border()` object that uses default values defined in flextable defaults formatting properties, i.e. default border color (see `set_flextable_defaults()`).

Usage

```
fp_border_default(
  color = flextable_global$defaults$border.color,
  style = "solid",
  width = 1
)
```

Arguments

color	border color - single character value (e.g. "#000000" or "black")
style	border style - single character value : "none" or "solid" or "dotted" or "dashed"
width	border width - an integer value : 0>= value

See Also

`hline()`, `vline()`

Other functions for defining formatting properties: `fp_text_default()`

Examples

```
fp_border_default(width = 2)
```

fp_text_default *Text formatting properties*

Description

Create a `fp_text()` object that uses default values defined in flextable defaults formatting properties, i.e. default font color, font size and font family (see `set_flextable_defaults()`). (see `set_flextable_defaults()`).

Usage

```
fp_text_default(  
  color = flextable_global$defaults$font.color,  
  font.size = flextable_global$defaults$font.size,  
  bold = FALSE,  
  italic = FALSE,  
  underlined = FALSE,  
  font.family = flextable_global$defaults$font.family,  
  vertical.align = "baseline",  
  shading.color = "transparent"  
)
```

Arguments

color	font color - a single character value specifying a valid color (e.g. "#000000" or "black").
font.size	font size (in point) - 0 or positive integer value.
bold	is bold
italic	is italic
underlined	is underlined
font.family	single character value specifying font name.
vertical.align	single character value specifying font vertical alignments. Expected value is one of the following : default 'baseline' or 'subscript' or 'superscript'
shading.color	shading color - a single character value specifying a valid color (e.g. "#000000" or "black").

See Also

[as_chunk\(\)](#)

Other functions for defining formatting properties: [fp_border_default\(\)](#)

Examples

```
fp_text_default(bold = TRUE)
```

get_flextable_defaults

Get flextable defaults formatting properties

Description

The current formatting properties are automatically applied to every flextable you produce. These default values are returned by this function.

Usage

```
get_flextable_defaults()
```

Value

a list containing default values.

See Also

Other functions related to themes: [set_flextable_defaults\(\)](#), [theme_alafoli\(\)](#), [theme_booktabs\(\)](#), [theme_box\(\)](#), [theme_tron_legacy\(\)](#), [theme_tron\(\)](#), [theme_vader\(\)](#), [theme_vanilla\(\)](#), [theme_zebra\(\)](#)

Examples

```
get_flextable_defaults()
```

gg_chunk

gg plots chunk wrapper

Description

This function is used to insert mini gg plots into flextable with function [compose\(\)](#). It should be used inside a call to [as_paragraph\(\)](#).

Usage

```
gg_chunk(value, width = 1, height = 0.2)
```

Arguments

value gg objects, stored in a list column.
width, height size of the resulting png file in inches

Note

This chunk option requires package `officedown` in a R Markdown context with Word output format. PowerPoint cannot mix images and text in a paragraph, images are removed when outputting to PowerPoint format.

See Also

Other chunk elements for paragraph: [as_bracket\(\)](#), [as_b\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_i\(\)](#), [as_sub\(\)](#), [as_sup\(\)](#), [hyperlink_text\(\)](#), [linerange\(\)](#), [lollipop\(\)](#), [minibar\(\)](#), [plot_chunk\(\)](#)

Examples

```

library(data.table)
library(flextable)
if(require("ggplot2")){
  my_cor_plot <- function(x){
    cols <- colnames(x)[sapply(x, is.numeric)]
    x <- x[, .SD, .SDcols = cols]
    cormat <- as.data.table(cor(x))
    cormat$var1 <- colnames(cormat)
    cormat <- melt(cormat, id.vars = "var1", measure.vars = cormat$var1,
                  variable.name = "var2", value.name = "correlation")
    ggplot(data = cormat, aes(x=var1, y=var2, fill=correlation)) +
      geom_tile() + coord_equal() +
      scale_fill_gradient2(low = "blue",
                           mid = "white", high = "red", limits = c(-1, 1),
                           guide = FALSE) + theme_void()
  }
  z <- as.data.table(iris)
  z <- z[, list(gg = list(my_cor_plot(.SD))), by = "Species"]
  ft <- flextable(z)
  ft <- mk_par(ft, j = "gg",
              value = as_paragraph(
                gg_chunk(value = gg, width = 1, height = 1)
              ))
  print(ft)
}

```

headers_flextable_at_bkm

add flextable at a bookmark location in document's header

Description

replace in the header of a document a paragraph containing a bookmark by a flextable. A bookmark will be considered as valid if enclosing words within a paragraph; i.e., a bookmark along two or more paragraphs is invalid, a bookmark set on a whole paragraph is also invalid, but bookmarking few words inside a paragraph is valid.

Usage

```
headers_flextable_at_bkm(x, bookmark, value)
```

Arguments

x	an rdocx object
bookmark	bookmark id
value	a flextable object

height	<i>Set flextable rows height</i>
--------	----------------------------------

Description

control rows height for a part of the flextable when the line height adjustment is "atleast" or "exact".

Usage

```
height(x, i = NULL, height, part = "body")
```

```
height_all(x, height, part = "all")
```

Arguments

x	flextable object
i	rows selection
height	height in inches
part	partname of the table

Illustrations**height_all**

height_all is a convenient function for setting the same height to all rows (selected with argument part).

Note

This function has no effect when the rule for line height is set to "auto" (see [hrule\(\)](#)), which is the default case, except with PowerPoint which does not support this automatic line height adjustment feature.

See Also

Other flextable dimensions: [autofit\(\)](#), [dim.flextable\(\)](#), [dim_pretty\(\)](#), [fit_to_width\(\)](#), [flextable_dim\(\)](#), [hrule\(\)](#), [set_table_properties\(\)](#), [width\(\)](#)

Examples

```
ft_1 <- flextable(head(iris))
ft_1 <- height(ft_1, height = .5)
ft_1 <- hrule(ft_1, rule = "exact")
ft_1
```

```
ft_2 <- flextable(head(iris))
ft_2 <- height_all(ft_2, height = 1)
ft_2 <- hrule(ft_2, rule = "exact")
ft_2
```

highlight	<i>Text Highlight Color</i>
-----------	-----------------------------

Description

change text highlight color of selected rows and columns of a flextable.

Usage

```
highlight(x, i = NULL, j = NULL, color = "yellow", part = "body", source = j)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
color	color to use as text highlighting color. If a function, function need to return a character vector of colors.
part	partname of the table (one of 'all', 'body', 'header', 'footer')
source	if color is a function, source is specifying the dataset column to be used as argument to color. This is only useful if j is colored with values contained in another (or other) column.

Illustrations

See Also

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [bold\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [font\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
my_color_fun <- function(x){
  out <- rep("yellow", length(x))
  out[x < quantile(x, .75)] <- "pink"
  out[x < quantile(x, .50)] <- "wheat"
  out[x < quantile(x, .25)] <- "gray90"
  out
}
ft <- flextable(head( mtcars, n = 10))
```

```
ft <- highlight(ft, j = "disp", i = ~ disp > 200, color = "yellow")
ft <- highlight(ft, j = ~ drat + wt + qsec, color = my_color_fun)
ft
```

hline	<i>set horizontal borders</i>
-------	-------------------------------

Description

The function is applying an horizontal border to inner content of one or all parts of a flextable. The lines are the bottom borders of selected cells.

Usage

```
hline(x, i = NULL, j = NULL, border = NULL, part = "body")
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner_v\(\)](#), [border_inner\(\)](#), [border_outer\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline_top\(\)](#), [vline_left\(\)](#), [vline_right\(\)](#), [vline\(\)](#)

Examples

```
library(officer)
std_border = fp_border(color="gray")

ft <- flextable(head(iris))
ft <- border_remove(x = ft)

# add horizontal borders
ft <- hline(ft, part="all", border = std_border )
ft
```

hline_bottom	<i>set bottom horizontal border</i>
--------------	-------------------------------------

Description

The function is applying an horizontal border to the bottom of one or all parts of a flextable. The line is the bottom border of selected parts.

Usage

```
hline_bottom(x, j = NULL, border = NULL, part = "body")
```

Arguments

x	a flextable object
j	columns selection
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner_v\(\)](#), [border_inner\(\)](#), [border_outer\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_top\(\)](#), [hline\(\)](#), [vline_left\(\)](#), [vline_right\(\)](#), [vline\(\)](#)

Examples

```
library(officer)
big_border = fp_border(color="orange", width = 3)

ft <- flextable(head(iris))
ft <- border_remove(x = ft)

# add/replace horizontal border on bottom
ft <- hline_bottom(ft, part="body", border = big_border )
ft
```

hline_top	<i>set top horizontal border</i>
-----------	----------------------------------

Description

The function is applying an horizontal border to the top of one or all parts of a flextable. The line is the top border of selected parts.

Usage

```
hline_top(x, j = NULL, border = NULL, part = "body")
```

Arguments

x	a flextable object
j	columns selection
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner_v\(\)](#), [border_inner\(\)](#), [border_outer\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline\(\)](#), [vline_left\(\)](#), [vline_right\(\)](#), [vline\(\)](#)

Examples

```
library(officer)
big_border = fp_border(color="orange", width = 3)

ft <- flextable(head(iris))
ft <- border_remove(x = ft)

# add horizontal border on top
ft <- hline_top(ft, part="all", border = big_border )
ft
```

hrule *Set flexible rule for rows heights*

Description

control rules of each height for a part of the flextable, this is only for Word and HTML outputs, it will not have any effect when output is PowerPoint.

Usage

```
hrule(x, i = NULL, rule = "auto", part = "body")
```

Arguments

x	flextable object
i	rows selection
rule	specify the meaning of the height. Possible values are "atleast" (height should be at least the value specified), "exact" (height should be exactly the value specified), or the default value "auto" (height is determined based on the height of the contents, so the value is ignored).
part	partname of the table, one of "all", "header", "body", "footer"

Illustrations

See Also

Other flextable dimensions: [autofit\(\)](#), [dim.flextable\(\)](#), [dim_pretty\(\)](#), [fit_to_width\(\)](#), [flextable_dim\(\)](#), [height\(\)](#), [set_table_properties\(\)](#), [width\(\)](#)

Examples

```
ft_1 <- flextable(head(iris))
ft_1 <- width(ft_1, width = 1.5)
ft_1 <- height(ft_1, height = 0.75, part = "header")
ft_1 <- hrule(ft_1, rule = "exact", part = "header")
ft_1

ft_2 <- hrule(ft_1, rule = "auto", part = "header")
ft_2
```

htmltools_value	<i>flextable as an HTML object</i>
-----------------	------------------------------------

Description

get a `div()` from a flextable object. This can be used in a shiny application. For an output within "R Markdown" document, use `knit_print.flextable`.

Usage

```
htmltools_value(x, ft.align = "center")
```

Arguments

<code>x</code>	a flextable object
<code>ft.align</code>	flextable alignment, supported values are 'left', 'center' and 'right'.

Value

an object marked as **HTML** ready to be used within a call to `shiny::renderUI` for example.

See Also

Other flextable print function: `as_raster()`, `flextable_to_rmd()`, `knit_print.flextable()`, `plot.flextable()`, `print.flextable()`, `save_as_docx()`, `save_as_html()`, `save_as_image()`, `save_as_pptx()`

Examples

```
htmltools_value(flextable(iris[1:5,]))
```

hyperlink_text	<i>chunk of text with hyperlink wrapper</i>
----------------	---

Description

The function lets add hyperlinks within flextable objects with function `compose()`. It should be used inside a call to `as_paragraph()`.

Usage

```
hyperlink_text(x, props = NULL, formatter = format_fun, url, ...)
```

Arguments

<code>x</code>	text or any element that can be formatted as text with function provided in argument <code>formatter</code> .
<code>props</code>	an <code>officer::fp_text()</code> object to be used to format the text. If not specified, it will be the default value corresponding to the cell.
<code>formatter</code>	a function that will format <code>x</code> as a character vector.
<code>url</code>	url to be used
<code>...</code>	additional arguments for <code>formatter</code> function.

Note

This chunk option requires package `officedown` in a R Markdown context with Word output format.

See Also

[compose\(\)](#)

Other chunk elements for paragraph: [as_bracket\(\)](#), [as_b\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_i\(\)](#), [as_sub\(\)](#), [as_sup\(\)](#), [gg_chunk\(\)](#), [linerange\(\)](#), [lollipop\(\)](#), [minibar\(\)](#), [plot_chunk\(\)](#)

Examples

```
dat <- data.frame(
  col = "Google it",
  href = "https://www.google.fr/search?source=hp&q=flextable+R+package",
  stringsAsFactors = FALSE)

ftab <- flextable(dat)
ftab <- compose( x = ftab, j = "col",
  value = as_paragraph(
    "This is a link: ",
    hyperlink_text(x = col, url = href ) ) )
ftab
```

italic

Set italic font

Description

change font decoration of selected rows and columns of a flextable.

Usage

```
italic(x, i = NULL, j = NULL, italic = TRUE, part = "body")
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
italic	boolean value
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations**See Also**

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [bold\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [font\(\)](#), [highlight\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
ft <- flextable(head(mtcars))
ft <- italic(ft, italic = TRUE, part = "header")
```

knit_print.flextable *Render flextable in rmarkdown*

Description

Function used to render flextable in knitr/rmarkdown documents.

You should not call this method directly. This function is used by the knitr package to automatically display a flextable in an "R Markdown" document from a chunk. However, it is recommended to read its documentation in order to get familiar with the different options available.

HTML, Word, PowerPoint and PDF outputs are supported.

Table captioning is a flextable feature compatible with R Markdown documents. The feature is available for HTML, PDF and Word documents. Compatibility with the "bookdown" package is also ensured, including the ability to produce captions so that they can be used in cross-referencing.

Usage

```
## S3 method for class 'flextable'
knit_print(x, ...)
```

Arguments

x	a flextable object
...	further arguments, not used.

Chunk options

Some features, often specific to an output format, are available to help you configure some global settings relative to the table output. knitr's chunk options are to be used to change the default settings:

chunk option	property	default value	HTML	d
ft.align	flextable alignment, supported values are 'left', 'center' and 'right'	'center'	yes	y
ft.shadow	HTML option, disable shadow dom (set to FALSE) for pagedown.	TRUE	yes	y
ft.split	Word option 'Allow row to break across pages' can be activated when TRUE.	FALSE	no	y
ft.tabcolsep	space between the text and the left/right border of its containing cell	8.0	no	
ft.arraystretch	height of each row relative to its default height	1.5	no	
ft.left	left coordinates in inches	1.0	no	
ft.top	top coordinates in inches	2.0	no	

Table caption

Captions can be defined in two ways.

The first is with the `set_caption` function. If it is used, the other method will be ignored. The second method is by using knitr chunk option `tab.cap`.

```
set_caption(x, caption = "my caption")
```

If `set_caption` function is not used, caption identifier will be read from knitr's chunk option `tab.id` or `label` if in a bookdown (this is to respect the bookdown standards).

```
tab.id='my_id' or label='my_id'
```

Word output provide more options such as ability to choose the prefix for numbering chunk for example. The table below expose these options:

chunk option	purpose	rmarkdown	bookdown
tab.cap.style	(Word only) style name to use for table captions	yes	yes
tab.cap.pre	(Word only) Prefix for numbering chunk (default to "Table")	yes	yes
tab.cap.sep	(Word only) Suffix for numbering chunk (default to ": ")	yes	yes
tab.cap	Caption label	yes	yes
tab.id	Caption reference unique identifier	yes	no
label	Caption reference unique identifier	no	yes

HTML output

HTML output is using shadow dom to encapsule the table into an isolated part of the page so that no clash happens with styles. Some output may not support this feature. To our knowledge, only the pagedown output is concerned. Use knitr chunk option `ft.shadow=FALSE` to disable shadow dom.

PDF output

Some features are not implemented in PDF due to technical infeasibility. These are the padding, `line_spacing` and height properties.

PowerPoint output

Auto-adjust Layout is not available for PowerPoint.

Also images cannot be integrated into tables with the PowerPoint format.

Note

Supported formats require some minimum **pandoc** versions:

Output format	pandoc minimal version
HTML	>= 1.12
Word (docx)	>= 2.0
PowerPoint (pptx)	>= 2.4
PDF	>= 1.12

See Also

Other flextable print function: [as_raster\(\)](#), [flextable_to_rmd\(\)](#), [htmltools_value\(\)](#), [plot.flextable\(\)](#), [print.flextable\(\)](#), [save_as_docx\(\)](#), [save_as_html\(\)](#), [save_as_image\(\)](#), [save_as_pptx\(\)](#)

Examples

```
# simple examples -----
demo_docx <- system.file(package = "flextable", "examples/rmd", "demo.Rmd")
rmd_file <- tempfile(fileext = ".Rmd")
file.copy(demo_docx, to = rmd_file, overwrite = TRUE)
rmd_file # R Markdown document used for demo
if(require("rmarkdown", quietly = TRUE)){
# knitr::opts_chunk$set(webshot = "webshot2")
# render(input = rmd_file, output_format = "word_document", output_file = "doc.docx")
# render(input = rmd_file, output_format = "pdf_document", output_file = "doc.pdf")
# render(input = rmd_file, output_format = "html_document", output_file = "doc.html")
# render(input = rmd_file, output_format = "powerpoint_presentation", output_file = "pres.pptx")
# render(input = rmd_file, output_format = "slidy_presentation", output_file = "slidy.html")
# render(input = rmd_file, output_format = "beamer_presentation", output_file = "beamer.pdf")
# render(input = rmd_file, output_format = "pagedown::html_paged", output_file = "paged.html")
}

## bookdown examples wth captions and cross ref -----
# captions_example <- system.file(
#   package = "flextable",
#   "examples/rmd", "captions_example.Rmd")
#
# dir_tmp <- tempfile(pattern = "dir")
# dir.create(dir_tmp, showWarnings = FALSE, recursive = TRUE)
# file.copy(captions_example, dir_tmp)
# rmd_file <- file.path(dir_tmp, basename(captions_example))
#
# file.copy(captions_example, to = rmd_file, overwrite = TRUE)
#
```



```

# if(require("rmarkdown", quietly = TRUE)){
#   render(input = rmd_file,
#           output_format = word_document(),
#           output_file = "doc.docx")
#   render(input = rmd_file,
#           output_format = pdf_document(latex_engine = "xelatex"),
#           output_file = "doc.pdf")
#   render(input = rmd_file,
#           output_format = html_document(),
#           output_file = "doc.html")
#
# # bookdown ----
# if(require("bookdown", quietly = TRUE)){
#   render(input = rmd_file, output_format = word_document2(),
#           output_file = "book.docx")
#   render(input = rmd_file,
#           output_format = pdf_document2(latex_engine = "xelatex"),
#           output_file = "book.pdf")
#   render(input = rmd_file,
#           output_format = html_document2(),
#           output_file = "book.html")
#
# # officedown ----
# if(require("officedown", quietly = TRUE)){
#   render(input = rmd_file,
#           output_format = markdown_document2(base_format=rdocx_document),
#           output_file = "officedown.docx")
#   }
# }
# }
# browseURL(dirname(rmd_file))

```

linerange

mini linerange chunk wrapper

Description

This function is used to insert lineranges into flextable with function `compose()`. It should be used inside a call to `as_paragraph()`

Usage

```

linerange(
  value,
  min = NULL,
  max = NULL,
  rangecol = "#CCCCCC",
  stickcol = "#FF0000",

```

```

    bg = "transparent",
    width = 1,
    height = 0.2,
    raster_width = 30
  )

```

Arguments

value	values containing the bar size
min	min bar size. Default min of value
max	max bar size. Default max of value
rangecol	bar color
stickcol	jauge color
bg	background color
width, height	size of the resulting png file in inches
raster_width	number of pixels used as width when interpolating value.

Note

This chunk option requires package `officedown` in a R Markdown context with Word output format. PowerPoint cannot mix images and text in a paragraph, images are removed when outputting to PowerPoint format.

See Also

[compose\(\)](#), [as_paragraph\(\)](#)

Other chunk elements for paragraph: [as_bracket\(\)](#), [as_b\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_i\(\)](#), [as_sub\(\)](#), [as_sup\(\)](#), [gg_chunk\(\)](#), [hyperlink_text\(\)](#), [lollipop\(\)](#), [minibar\(\)](#), [plot_chunk\(\)](#)

Examples

```

myft <- flextable( head(iris, n = 10 ))

myft <- compose( myft, j = 1,
  value = as_paragraph(
    linerange(value = Sepal.Length)
  ),
  part = "body")

autofit(myft)

```

line_spacing	<i>Set text alignment</i>
--------------	---------------------------

Description

change text alignment of selected rows and columns of a flextable.

Usage

```
line_spacing(x, i = NULL, j = NULL, space = 1, part = "body")
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
space	space between lines of text, 1 is single line spacing, 2 is double line spacing.
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations**See Also**

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [bold\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [font\(\)](#), [highlight\(\)](#), [italic\(\)](#), [padding\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
ft <- flextable(head(mtcars)[,3:6])
ft <- line_spacing(ft, space = 1.6, part = "all")
ft <- set_table_properties(ft, layout = "autofit")
ft
```

lollipop	<i>mini lollipop chart chunk wrapper</i>
----------	--

Description

This function is used to insert lollipop charts into flextable with function [compose\(\)](#). It should be used inside a call to [as_paragraph\(\)](#)

Usage

```

lollipop(
  value,
  min = NULL,
  max = NULL,
  rangecol = "#CCCCCC",
  bg = "transparent",
  width = 1,
  height = 0.2,
  raster_width = 30,
  positivecol = "#00CC00",
  negativecol = "#CC0000",
  neutralcol = "#CCCCCC",
  neutralrange = c(0, 0),
  rectanglesize = 2
)

```

Arguments

value	values containing the bar size
min	min bar size. Default min of value
max	max bar size. Default max of value
rangecol	bar color
bg	background color
width, height	size of the resulting png file in inches
raster_width	number of pixels used as width
positivecol	box color of positive values
negativecol	box color of negative values
neutralcol	box color of neutral values
neutralrange	minimal and maximal range of neutral values (default: 0)
rectanglesize	size of the rectangle (default: 2, max: 5) when interpolating value.

Illustrations**Note**

This chunk option requires package `officedown` in a R Markdown context with Word output format. PowerPoint cannot mix images and text in a paragraph, images are removed when outputting to PowerPoint format.

See Also

[compose\(\)](#), [as_paragraph\(\)](#)

Other chunk elements for paragraph: [as_bracket\(\)](#), [as_b\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_i\(\)](#), [as_sub\(\)](#), [as_sup\(\)](#), [gg_chunk\(\)](#), [hyperlink_text\(\)](#), [linerange\(\)](#), [minibar\(\)](#), [plot_chunk\(\)](#)

Examples

```
iris$Sepal.Ratio <- (iris$Sepal.Length - mean(iris$Sepal.Length))/mean(iris$Sepal.Length)
ft <- flextable( tail(iris, n = 10 ) )

ft <- compose( ft, j = "Sepal.Ratio", value = as_paragraph(
  lollipop(value = Sepal.Ratio, min=-.25, max=.25)
),
part = "body")

ft <- autofit(ft)
ft
```

merge_at

Merge flextable cells into a single one

Description

Merge flextable cells into a single one. All rows and columns must be consecutive.

Usage

```
merge_at(x, i = NULL, j = NULL, part = "body")
```

Arguments

x	flextable object
i, j	columns and rows to merge
part	partname of the table where merge has to be done.

See Also

Other flextable merging function: [merge_h_range\(\)](#), [merge_h\(\)](#), [merge_none\(\)](#), [merge_v\(\)](#)

Examples

```
ft_merge <- flextable( head( mtcars ), cwidth = .5 )
ft_merge <- merge_at( ft_merge, i = 1:2, j = 1:2 )
ft_merge
```

merge_h	<i>Merge flextable cells horizontally</i>
---------	---

Description

Merge flextable cells horizontally when consecutive cells have identical values. Text of formatted values are used to compare values.

Usage

```
merge_h(x, i = NULL, part = "body")
```

Arguments

x	flextable object
i	rows where cells have to be merged.
part	partname of the table where merge has to be done.

See Also

Other flextable merging function: [merge_at\(\)](#), [merge_h_range\(\)](#), [merge_none\(\)](#), [merge_v\(\)](#)

Examples

```
dummy_df <- data.frame( col1 = letters,
  col2 = letters, stringsAsFactors = FALSE )
ft_merge <- flextable(dummy_df)
ft_merge <- merge_h(x = ft_merge)
ft_merge
```

merge_h_range	<i>rowwise merge of a range of columns</i>
---------------	--

Description

Merge flextable columns into a single one for each selected rows. All columns must be consecutive.

Usage

```
merge_h_range(x, i = NULL, j1 = NULL, j2 = NULL, part = "body")
```

Arguments

x	flextable object
i	selected rows
j1, j2	selected columns that will define the range of columns to merge.
part	partname of the table where merge has to be done.

Illustrations**See Also**

Other flextable merging function: [merge_at\(\)](#), [merge_h\(\)](#), [merge_none\(\)](#), [merge_v\(\)](#)

Examples

```
ft <- flextable( head( mtcars ), cwidth = .5 )
ft <- theme_box( ft )
ft <- merge_h_range( ft, i = ~ cyl == 6, j1 = "am", j2 = "carb")
ft <- flextable::align( ft, i = ~ cyl == 6, align = "center")
ft
```

 merge_none

Delete flextable merging informations

Description

Delete all merging informations from a flextable.

Usage

```
merge_none(x, part = "all")
```

Arguments

x	flextable object
part	partname of the table where merge has to be done.

Illustrations**See Also**

Other flextable merging function: [merge_at\(\)](#), [merge_h_range\(\)](#), [merge_h\(\)](#), [merge_v\(\)](#)

Examples

```
typology <- data.frame(
  col_keys = c( "Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", "Species" ),
  what = c("Sepal", "Sepal", "Petal", "Petal", "Species"),
  measure = c("Length", "Width", "Length", "Width", "Species"),
  stringsAsFactors = FALSE )

ft <- flextable( head( iris ) )
```

```
ft <- set_header_df(ft, mapping = typology, key = "col_keys" )
ft <- merge_v(ft, j = c("Species"))

ft <- theme_tron_legacy( merge_none( ft ) )
ft
```

merge_v

Merge flextable cells vertically

Description

Merge flextable cells vertically when consecutive cells have identical values. Text of formatted values are used to compare values if available.

Two options are available, either a column-by-column algorithm or an algorithm where the combinations of these columns are used once for all target columns.

Usage

```
merge_v(x, j = NULL, target = NULL, part = "body", combine = FALSE)
```

Arguments

x	flextable object
j	column to used to find consecutive values to be merged. Columns from original dataset can also be used.
target	columns names where cells have to be merged.
part	partname of the table where merge has to be done.
combine	If the value is TRUE, the columns defined by j will be combined into a single column/value and the consecutive values of this result will be used. Otherwise, the columns are inspected one by one to perform cell merges.

Illustrations

See Also

Other flextable merging function: [merge_at\(\)](#), [merge_h_range\(\)](#), [merge_h\(\)](#), [merge_none\(\)](#)

Examples

```
ft_merge <- flextable(mtcars)
ft_merge <- merge_v(ft_merge, j = c("gear", "carb"))
ft_merge

data_ex <- structure(list(srdr_id = c(
  "175124", "175124", "172525", "172525",
```



```

      "172545", "172545", "172609", "172609", "172609"
    ), substances = c(
      "alcohol",
      "alcohol", "alcohol", "alcohol", "cannabis",
      "cannabis", "alcohol\n cannabis\n other drugs",
      "alcohol\n cannabis\n other drugs",
      "alcohol\n cannabis\n other drugs"
    ), full_name = c(
      "TAU", "MI", "TAU", "MI (parent)", "TAU", "MI",
      "TAU", "MI", "MI"
    ), article_arm_name = c(
      "Control", "WISEteens",
      "Treatment as usual", "Brief MI (b-MI)", "Assessed control",
      "Intervention", "Control", "Computer BI", "Therapist BI"
    )), row.names = c(
      NA,
      -9L
    ), class = c("tbl_df", "tbl", "data.frame"))

ft_1 <- flextable(data_ex)
ft_1 <- theme_box(ft_1)
ft_2 <- merge_v(ft_1, j = "srd_r_id",
  target = c("srd_r_id", "substances"))
ft_2

```

minibar

mini barplots chunk wrapper

Description

This function is used to insert bars into flextable with function `compose()`. It should be used inside a call to `as_paragraph()`

Usage

```

minibar(
  value,
  max = NULL,
  barcol = "#CCCCCC",
  bg = "transparent",
  width = 1,
  height = 0.2
)

```

Arguments

value	values containing the bar size
max	max bar size

barcol	bar color
bg	background color
width, height	size of the resulting png file in inches

Illustrations

Note

This chunk option requires package `officedown` in a R Markdown context with Word output format. PowerPoint cannot mix images and text in a paragraph, images are removed when outputting to PowerPoint format.

See Also

[compose\(\)](#), [as_paragraph\(\)](#)

Other chunk elements for paragraph: [as_bracket\(\)](#), [as_b\(\)](#), [as_chunk\(\)](#), [as_image\(\)](#), [as_i\(\)](#), [as_sub\(\)](#), [as_sup\(\)](#), [gg_chunk\(\)](#), [hyperlink_text\(\)](#), [linerange\(\)](#), [lollipop\(\)](#), [plot_chunk\(\)](#)

Examples

```
ft <- flextable( head(iris, n = 10 ))

ft <- compose(ft, j = 1,
  value = as_paragraph(
    minibar(value = Sepal.Length, max = max(Sepal.Length))
  ),
  part = "body")

ft <- autofit(ft)
ft
```

padding	<i>Set paragraph paddings</i>
---------	-------------------------------

Description

change paddings of selected rows and columns of a flextable.

Usage

```
padding(
  x,
  i = NULL,
  j = NULL,
  padding = NULL,
```

```
padding.top = NULL,
padding.bottom = NULL,
padding.left = NULL,
padding.right = NULL,
part = "body"
)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
padding	padding (shortcut for top, bottom, left and right)
padding.top	padding top
padding.bottom	padding bottom
padding.left	padding left
padding.right	padding right
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [bold\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [font\(\)](#), [highlight\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [rotate\(\)](#), [valign\(\)](#)

Examples

```
ft_1 <- flextable(head(iris))
ft_1 <- theme_vader(ft_1)
ft_1 <- padding(ft_1, padding.top = 4, part = "all")
ft_1 <- padding(ft_1, j = 1, padding.right = 40)
ft_1 <- padding(ft_1, i = 3, padding.top = 40)
ft_1 <- padding(ft_1, padding.top = 10, part = "header")
ft_1 <- padding(ft_1, padding.bottom = 10, part = "header")
ft_1 <- autofit(ft_1)
ft_1
```

ph_with.flextable *add a flextable into a PowerPoint slide*

Description

Add a flextable in a PowerPoint document object produced by `officer::read_pptx()`.

Usage

```
## S3 method for class 'flextable'  
ph_with(x, value, location, ...)
```

Arguments

x	a pptx device
value	flextable object
location	a location for a placeholder. See <code>officer::ph_location_type()</code> for example.
...	unused arguments.

Note

The width and height of the table can not be set with `location`. Use functions `width()`, `height()`, `autofit()` and `dim_pretty()` instead. The overall size is resulting from cells, paragraphs and text properties (i.e. padding, font size, border widths).

Examples

```
library(officer)  
  
ft = flextable(head(iris))  
  
doc <- read_pptx()  
doc <- add_slide(doc, "Title and Content", "Office Theme")  
doc <- ph_with(doc, ft, location = ph_location_left())  
  
fileout <- tempfile(fileext = ".pptx")  
print(doc, target = fileout)
```

plot.flextable	<i>plot a flextable</i>
----------------	-------------------------

Description

save a flextable as an image and display the result in a new R graphics window.

Usage

```
## S3 method for class 'flextable'  
plot(x, zoom = 2, expand = 2, ...)
```

Arguments

x	a flextable object
zoom, expand	parameters used by webshot function.
...	additional parameters sent to as_raster() function

Note

This function requires packages: webshot and magick.

See Also

Other flextable print function: [as_raster\(\)](#), [flextable_to_rmd\(\)](#), [htmltools_value\(\)](#), [knit_print.flextable\(\)](#), [print.flextable\(\)](#), [save_as_docx\(\)](#), [save_as_html\(\)](#), [save_as_image\(\)](#), [save_as_pptx\(\)](#)

Examples

```
ftab <- flextable( head( mtcars ) )  
ftab <- autofit(ftab)  
## Not run:  
if( require("webshot") ){  
  plot(ftab)  
}  
  
## End(Not run)
```

plot_chunk	<i>mini plots chunk wrapper</i>
------------	---------------------------------

Description

This function is used to insert mini plots into flextable with function `compose()`. It should be used inside a call to `as_paragraph()`.

Available plots are 'box', 'line', 'points', 'density'.

Usage

```
plot_chunk(
  value,
  width = 1,
  height = 0.2,
  type = "box",
  free_scale = FALSE,
  ...
)
```

Arguments

value	a numeric vector, stored in a list column.
width, height	size of the resulting png file in inches
type	type of the plot: 'box', 'line', 'points' or 'density'.
free_scale	Should scales be free (TRUE or FALSE, the default value).
...	arguments sent to plot functions (see <code>par()</code>)

Note

This chunk option requires package `officedown` in a R Markdown context with Word output format. PowerPoint cannot mix images and text in a paragraph, images are removed when outputting to PowerPoint format.

See Also

Other chunk elements for paragraph: `as_bracket()`, `as_b()`, `as_chunk()`, `as_image()`, `as_i()`, `as_sub()`, `as_sup()`, `gg_chunk()`, `hyperlink_text()`, `linorange()`, `lollipop()`, `minibar()`

Examples

```
library(data.table)
library(flextable)

z <- as.data.table(iris)
z <- z[, list(
```

```

  Sepal.Length = mean(Sepal.Length, na.rm = TRUE),
  z = list(.SD$Sepal.Length)
), by = "Species"]

ft <- flextable(z,
  col_keys = c("Species", "Sepal.Length", "box", "density"))
ft <- mk_par(ft, j = "box", value = as_paragraph(
  plot_chunk(value = z, type = "box",
    border = "red", col = "transparent")))
ft <- mk_par(ft, j = "density", value = as_paragraph(
  plot_chunk(value = z, type = "dens", col = "red")))
ft <- set_table_properties(ft, layout = "autofit", width = .6)
ft <- set_header_labels(ft, box = "boxplot", density = "density")
theme_vanilla(ft)

```

print.flextable *flextable printing*

Description

print a flextable object to format html, docx, pptx or as text (not for display but for informative purpose). This function is to be used in an interactive context.

Usage

```

## S3 method for class 'flextable'
print(x, preview = "html", ...)

```

Arguments

x	flextable object
preview	preview type, one of c("html", "pptx", "docx", "log"). When "log" is used, a description of the flextable is printed.
...	unused argument

Note

When argument preview is set to "docx" or "pptx", an external client linked to these formats (Office is installed) is used to edit a document. The document is saved in the temporary directory of the R session and will be removed when R session will be ended.

When argument preview is set to "html", an external client linked to these HTML format is used to display the table. If RStudio is used, the Viewer is used to display the table.

Note also that a print method is used when flextable are used within R markdown documents. See [knit_print.flextable\(\)](#).

See Also

Other flextable print function: [as_raster\(\)](#), [flextable_to_rmd\(\)](#), [htmltools_value\(\)](#), [knit_print.flextable\(\)](#), [plot.flextable\(\)](#), [save_as_docx\(\)](#), [save_as_html\(\)](#), [save_as_image\(\)](#), [save_as_pptx\(\)](#)

proc_freq	<i>frequency table as flextable</i>
-----------	-------------------------------------

Description

This function compute a two way contingency table and make a flextable with the result.

Usage

```
proc_freq(  
  x,  
  row,  
  col,  
  main = "",  
  include.row_percent = TRUE,  
  include.column_percent = TRUE,  
  include.table_percent = TRUE,  
  include.column_total = TRUE,  
  include.row_total = TRUE,  
  include.header_row = TRUE,  
  weight = NULL  
)
```

Arguments

x	data.frame object
row	character column names for row
col	character column names for column
main	character title
include.row_percent	boolean whether to include the row percents; defaults to TRUE
include.column_percent	boolean whether to include the column percents; defaults to TRUE
include.table_percent	boolean whether to include the table percents; defaults to TRUE
include.column_total	boolean whether to include the row of column totals; defaults to TRUE
include.row_total	boolean whether to include the column of row totals; defaults to TRUE
include.header_row	boolean whether to include the header row; defaults to TRUE
weight	character column name for weight

Author(s)

Titouan Robert

Examples

```
proc_freq(mtcars, "vs", "gear")
proc_freq(mtcars, "gear", "vs")
proc_freq(mtcars, "gear", "vs", weight = "wt")
proc_freq(mtcars, "gear", "vs", "My title")
```

rotate	<i>rotate cell text</i>
--------	-------------------------

Description

apply a rotation to cell text. The text direction can be "lrb" which mean from left to right and top to bottom (the default direction). In some cases, it can be useful to be able to change the direction, when the table headers are huge for example, header labels can be rendered as "tblr" (top to bottom and right to left) corresponding to a 90 degrees rotation or "btlr" corresponding to a 270 degrees rotation.

Usage

```
rotate(x, i = NULL, j = NULL, rotation, align = "center", part = "body")
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
rotation	one of "lrb", "tblr", "btlr". Note that "btlr" is ignored when output is HTML.
align	vertical alignment of paragraph within cell, one of "center" or "top" or "bottom".
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Details

When function `autofit` is used, the rotation will be ignored. In that case, use `dim_pretty` and `width` instead of `autofit`.

Illustrations

See Also

Other sugar functions for table style: `align()`, `bg()`, `bold()`, `color()`, `empty_blanks()`, `fontsize()`, `font()`, `highlight()`, `italic()`, `line_spacing()`, `padding()`, `valign()`

Examples

```

library(flextable)

ft <- flextable(head(iris))

# measure column widths but only for the body part
w_body <- dim_pretty(ft, part = "body")$widths
# measure column widths only for the header part and get the max
# as height value for rotated text
h_header <- max( dim_pretty(ft, part = "header")$widths )

ft <- rotate(ft, j = 1:4, rotation="btlr",part="header")
ft <- rotate(ft, j = 5, rotation="tbrl",part="header")

ft <- valign(ft, valign = "center", part = "header")
ft <- flextable::align(ft, align = "center", part = "all")

# Manage header height
ft <- height(ft, height = h_header * 1.1, part = "header")
# ... mainly because Word don't handle auto height with rotated headers
ft <- hrule(ft, i = 1, rule = "exact", part = "header")

ft

```

save_as_docx

save flextable objects in an Word file

Description

sugar function to save flextable objects in an Word file.

Usage

```
save_as_docx(..., values = NULL, path, pr_section = NULL)
```

Arguments

...	flextable objects, objects, possibly named. If named objects, names are used as titles.
values	a list (possibly named), each element is a flextable object. If named objects, names are used as titles. If provided, argument ... will be ignored.
path	Word file to be created
pr_section	a prop_section object that can be used to define page layout such as orientation, width and height.

See Also

Other flextable print function: [as_raster\(\)](#), [flextable_to_rmd\(\)](#), [htmltools_value\(\)](#), [knit_print.flextable\(\)](#), [plot.flextable\(\)](#), [print.flextable\(\)](#), [save_as_html\(\)](#), [save_as_image\(\)](#), [save_as_pptx\(\)](#)

Examples

```
tf <- tempfile(fileext = ".docx")

library(officer)
ft1 <- flextable( head( iris ) )
save_as_docx(ft1, path = tf)

ft2 <- flextable( head( mtcars ) )
sect_properties <- prop_section(
  page_size = page_size(orient = "landscape",
    width = 8.3, height = 11.7),
  type = "continuous",
  page_margins = page_mar()
)
save_as_docx(`iris table` = ft1, `mtcars table` = ft2,
  path = tf, pr_section = sect_properties)
```

save_as_html

Save a Flextable in an HTML File

Description

save a flextable in an HTML file. This function is useful to save the flextable in HTML file without using R Markdown (it is highly recommended to use R Markdown instead).

Usage

```
save_as_html(
  ...,
  values = NULL,
  path,
  encoding = "utf-8",
  title = deparse(sys.call())
)
```

Arguments

...	flextable objects, objects, possibly named. If named objects, names are used as titles.
values	a list (possibly named), each element is a flextable object. If named objects, names are used as titles. If provided, argument ... will be ignored.
path	HTML file to be created
encoding	encoding to be used in the HTML file
title	page title

See Also

Other flextable print function: `as_raster()`, `flextable_to_rmd()`, `htmltools_value()`, `knit_print.flextable()`, `plot.flextable()`, `print.flextable()`, `save_as_docx()`, `save_as_image()`, `save_as_pptx()`

Examples

```
ft1 <- flextable( head( iris ) )
tf1 <- tempfile(fileext = ".html")
save_as_html(ft1, path = tf1)
# browseURL(tf1)
```

```
ft2 <- flextable( head( mtcars ) )
tf2 <- tempfile(fileext = ".html")
save_as_html(
  `iris table` = ft1,
  `mtcars table` = ft2,
  path = tf2,
  title = "rho000")
# browseURL(tf2)
```

save_as_image

save a flextable as an image

Description

save a flextable as a png, pdf or jpeg image.

Image generated with package 'webshot' or package 'webshot2'. **Package 'webshot2' should be preferred** as 'webshot' can have issues with some properties (i.e. bold are not rendered for some users).

Usage

```
save_as_image(x, path, zoom = 3, expand = 10, webshot = "webshot")
```

Arguments

x	a flextable object
path	image file to be created. It should end with .png, .pdf, or .jpeg.
zoom, expand	parameters used by webshot function.
webshot	webshot package as a scalar character, one of "webshot" or "webshot2".

Note

This function requires package webshot or webshot2.

See Also

Other flextable print function: `as_raster()`, `flextable_to_rmd()`, `htmltools_value()`, `knit_print.flextable()`, `plot.flextable()`, `print.flextable()`, `save_as_docx()`, `save_as_html()`, `save_as_pptx()`

Examples

```
ft <- flextable( head( mtcars ) )
ft <- autofit(ft)
tf <- tempfile(fileext = ".png")
## Not run:
if( require("webshot") ){
  save_as_image(x = ft, path = "myimage.png")
}

## End(Not run)
```

 save_as_pptx

save flextable objects in an PowerPoint file

Description

sugar function to save flextable objects in an PowerPoint file.

Usage

```
save_as_pptx(..., values = NULL, path)
```

Arguments

...	flextable objects, objects, possibly named. If named objects, names are used as slide titles.
values	a list (possibly named), each element is a flextable object. If named objects, names are used as slide titles. If provided, argument ... will be ignored.
path	PowerPoint file to be created

See Also

Other flextable print function: `as_raster()`, `flextable_to_rmd()`, `htmltools_value()`, `knit_print.flextable()`, `plot.flextable()`, `print.flextable()`, `save_as_docx()`, `save_as_html()`, `save_as_image()`

Examples

```
ft1 <- flextable( head( iris ) )
tf <- tempfile(fileext = ".pptx")
save_as_pptx(ft1, path = tf)

ft2 <- flextable( head( mtcars ) )
tf <- tempfile(fileext = ".pptx")
save_as_pptx(`iris table` = ft1, `mtcars table` = ft2, path = tf)
```

set_caption	<i>Set Caption</i>
-------------	--------------------

Description

Set caption value in a flextable.

- The caption will be associated with a paragraph style when the output is Word. Also it can be numbered as a auto-numbered Word computed value.
- In HTML and PDF, the style of the caption is managed at the document level and not with flextable.
- The PowerPoint format ignores captions.

Usage

```
set_caption(  
    x,  
    caption,  
    autonum = NULL,  
    style = "Table Caption",  
    html_escape = TRUE  
)
```

Arguments

x	flextable object
caption	caption value
autonum	an autonum representation. See officer::run_autonum() . This has only an effect when output is Word. If used, the caption is preceded by an auto-number sequence. In this case, the caption is preceded by an auto-number sequence that can be cross referenced.
style	caption paragraph style name. These names are available with function officer::styles_info() when output is Word; if HTML, the value is set as class value in the caption tag.
html_escape	should HTML entities be escaped so that it can be safely included as text or an attribute value within an HTML document.

See Also

[flextable\(\)](#)

Examples

```
ftab <- flextable( head( iris ) )
ftab <- set_caption(ftab, "my caption")
ftab

library(officer)
autonum <- run_autonum(seq_id = "tab", bkm = "mtcars")
ftab <- flextable( head( mtcars ) )
ftab <- set_caption(ftab, caption = "mtcars data", autonum = autonum)
ftab
```

set_flextable_defaults

Modify flextable defaults formatting properties

Description

The current formatting properties (see [get_flextable_defaults\(\)](#)) are automatically applied to every flextable you produce. Use `set_flextable_defaults()` to override them.

Usage

```
set_flextable_defaults(
  font.family = NULL,
  font.size = NULL,
  font.color = NULL,
  text.align = NULL,
  padding.bottom = NULL,
  padding.top = NULL,
  padding.left = NULL,
  padding.right = NULL,
  border.color = NULL,
  background.color = NULL,
  table.layout = NULL,
  decimal.mark = NULL,
  big.mark = NULL,
  digits = NULL,
  na_str = NULL,
  fmt_date = NULL,
  fmt_datetime = NULL,
  extra_css = NULL,
  fonts_ignore = NULL,
  theme_fun = NULL,
  post_process_pdf = NULL,
  post_process_docx = NULL,
  post_process_html = NULL,
  post_process_pptx = NULL
)
```

Arguments

font.family	single character value specifying font name.
font.size	font size (in point) - 0 or positive integer value.
font.color	font color - a single character value specifying a valid color (e.g. "#000000" or "black").
text.align	text alignment - a single character value, expected value is one of 'left', 'right', 'center', 'justify'.
padding.bottom, padding.top, padding.left, padding.right	paragraph paddings - 0 or positive integer value.
border.color	border color - single character value (e.g. "#000000" or "black").
background.color	cell background color - a single character value specifying a valid color (e.g. "#000000" or "black").
table.layout	'autofit' or 'fixed' algorithm. Default to 'autofit'.
decimal.mark, big.mark, digits, na_str	formatC arguments used by colformat_num() and colformat_int() .
fmt_date, fmt_datetime	formats for date and datetime columns as documented in strptime() . Default to '%Y-%m-%d' and '%Y-%m-%d %H:%M:%S'.
extra_css	css instructions to be integrated with the table. By default, it contains only the style for captions.
fonts_ignore	if TRUE, pdf-engine pdflatex can be used instead of xelatex or lualatex. If pdflatex is used, fonts will be ignored because they are not supported by pdflatex, whereas with the xelatex and lualatex engines they are.
theme_fun	a single character value (the name of the theme function to be applied) or a theme function (input is a flextable, output is a flextable).
post_process_pdf, post_process_docx, post_process_html, post_process_pptx	Post-processing functions that will allow you to customize the display by output type (pdf, html, docx, pptx). They are executed just before printing the table.

Value

a list containing previous default values.

Illustrations**See Also**

Other functions related to themes: [get_flextable_defaults\(\)](#), [theme_alafoli\(\)](#), [theme_booktabs\(\)](#), [theme_box\(\)](#), [theme_tron_legacy\(\)](#), [theme_tron\(\)](#), [theme_vader\(\)](#), [theme_vanilla\(\)](#), [theme_zebra\(\)](#)

Examples

```
ft_1 <- qflexible(head(airquality))
ft_1

old <- set_flexible_defaults(
  font.color = "#AA8855",
  border.color = "#8855AA")
ft_2 <- qflexible(head(airquality))
ft_2

do.call(set_flexible_defaults, old)
```

set_formatter *set column formatter functions*

Description

Define formatter functions associated to each column key. Functions have a single argument (the vector) and are returning the formatted values as a character vector.

Usage

```
set_formatter(x, ..., values = NULL, part = "body")

set_formatter_type(
  x,
  fmt_double = "%.03f",
  fmt_integer = "%.0f",
  fmt_date = "%Y-%m-%d",
  fmt_datetime = "%Y-%m-%d %H:%M:%S",
  true = "true",
  false = "false",
  na_str = ""
)
```

Arguments

x	a flextable object
...	Name-value pairs of functions, names should be existing col_key values
values	a list of name-value pairs of functions, names should be existing col_key values. If values is supplied argument ... is ignored.
part	partname of the table (one of 'body' or 'header' or 'footer')
fmt_double, fmt_integer	arguments used by sprintf to format double and integer columns.
fmt_date, fmt_datetime	arguments used by format to format date and date time columns.
false, true	string to be used for logical columns
na_str	string for NA values

Illustrations

set_formatter_type

set_formatter_type is an helper function to quickly define formatter functions regarding to column types.

This function will be deprecated in favor of the colformat_* functions, for example `colformat_double()`.

See Also

Other cells formatters: `colformat_char()`, `colformat_datetime()`, `colformat_date()`, `colformat_double()`, `colformat_image()`, `colformat_int()`, `colformat_lgl()`, `colformat_num()`, `compose()`

Other cells formatters: `colformat_char()`, `colformat_datetime()`, `colformat_date()`, `colformat_double()`, `colformat_image()`, `colformat_int()`, `colformat_lgl()`, `colformat_num()`, `compose()`

Examples

```
ft <- flextable( head( iris ) )
ft <- set_formatter( x = ft,
  Sepal.Length = function(x) sprintf("%.02f", x),
  Sepal.Width = function(x) sprintf("%.04f", x)
)
ft <- theme_vanilla( ft )
ft
```

set_header_footer_df *Set flextable's header or footer rows*

Description

Use a data.frame to specify flextable's header or footer rows.

The data.frame must contain a column whose values match flextable `col_keys` argument, this column will be used as join key. The other columns will be displayed as header or footer rows. The leftmost column is used as the top header/footer row and the rightmost column is used as the bottom header/footer row.

Usage

```
set_header_df(x, mapping = NULL, key = "col_keys")
```

```
set_footer_df(x, mapping = NULL, key = "col_keys")
```

Arguments

x	a flextable object
mapping	a data.frame specifying for each colname content of the column.
key	column to use as key when joining data_mapping.

Illustrations**See Also**

Other headers and footers: [add_header_lines\(\)](#), [add_header_row\(\)](#), [add_header\(\)](#), [set_header_labels\(\)](#)

Examples

```

typology <- data.frame(
  col_keys = c( "Sepal.Length", "Sepal.Width", "Petal.Length",
               "Petal.Width", "Species" ),
  what = c("Sepal", "Sepal", "Petal", "Petal", "Species"),
  measure = c("Length", "Width", "Length", "Width", "Species"),
  stringsAsFactors = FALSE )

ft_1 <- flextable( head( iris ))
ft_1 <- set_header_df(ft_1, mapping = typology, key = "col_keys" )
ft_1 <- merge_h(ft_1, part = "header")
ft_1 <- merge_v(ft_1, j = "Species", part = "header")
ft_1 <- theme_vanilla(ft_1)
ft_1 <- fix_border_issues(ft_1)
ft_1

typology <- data.frame(
  col_keys = c( "Sepal.Length", "Sepal.Width", "Petal.Length",
               "Petal.Width", "Species" ),
  unit = c("(cm)", "(cm)", "(cm)", "(cm)", "" ),
  stringsAsFactors = FALSE )
ft_2 <- set_footer_df(ft_1, mapping = typology, key = "col_keys" )
ft_2 <- italic(ft_2, italic = TRUE, part = "footer" )
ft_2 <- theme_booktabs(ft_2)
ft_2 <- fix_border_issues(ft_2)
ft_2

```

set_header_labels *Set flextable's headers labels*

Description

This function set labels for specified columns in a single row header of a flextable.

Usage

```
set_header_labels(x, ..., values = NULL)
```

Arguments

x	a flextable object
...	named arguments (names are data colnames), each element is a single character value specifying label to use.
values	a named list (names are data colnames), each element is a single character value specifying label to use. If provided, argument ... will be ignored.

Illustrations**See Also**

Other headers and footers: [add_header_lines\(\)](#), [add_header_row\(\)](#), [add_header\(\)](#), [set_header_footer_df](#)

Examples

```
ft <- flextable( head( iris ))
ft <- set_header_labels(ft, Sepal.Length = "Sepal length",
  Sepal.Width = "Sepal width", Petal.Length = "Petal length",
  Petal.Width = "Petal width"
)

ft <- flextable( head( iris ))
ft <- set_header_labels(ft,
  values = list(Sepal.Length = "Sepal length",
    Sepal.Width = "Sepal width",
    Petal.Length = "Petal length",
    Petal.Width = "Petal width" ) )
ft
```

set_table_properties *Global table properties*

Description

Set table layout and table width. Default to fixed algorithm.

If layout is fixed, column widths will be used to display the table; width is ignored.

If layout is autofit, column widths will not be used; table width is used (as a percentage).

Usage

```
set_table_properties(x, layout = "fixed", width = 1)
```

Arguments

x	flextable object
layout	'autofit' or 'fixed' algorithm. Default to 'autofit'.
width	The parameter has a different effect depending on the output format. In HTML, it is the width of the space that the table should occupy. In Word, it is a preferred size and Word may decide not to strictly stick to it. It has no effect on PowerPoint and PDF output.

Illustrations**Note**

PowerPoint output ignore autofit layout as this algorithm does not exist for this Microsoft format.

See Also

Other flextable dimensions: [autofit\(\)](#), [dim.flextable\(\)](#), [dim_pretty\(\)](#), [fit_to_width\(\)](#), [flextable_dim\(\)](#), [height\(\)](#), [hrule\(\)](#), [width\(\)](#)

Examples

```
library(flextable)
ft_1 <- qflextable(head(cars))
ft_2 <- set_table_properties(ft_1, width = .5, layout = "autofit")
ft_2
```

style

Set flextable style

Description

Modify flextable text, paragraphs and cells formatting properties. It allows to specify a set of formatting properties for a selection instead of using multiple functions (i.e bold, italic, bg) that should all be applied to the same selection of rows and columns.

Usage

```
style(
  x,
  i = NULL,
  j = NULL,
  pr_t = NULL,
  pr_p = NULL,
  pr_c = NULL,
  part = "body"
)
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
pr_t	object(s) of class fp_text
pr_p	object(s) of class fp_par
pr_c	object(s) of class fp_cell
part	partname of the table (one of 'all', 'body', 'header' or 'footer')

Illustrations**Examples**

```
library(officer)
def_cell <- fp_cell(border = fp_border(color="wheat"))

def_par <- fp_par(text.align = "center")

ft <- flextable(head(mtcars))

ft <- style( ft, pr_c = def_cell, pr_p = def_par, part = "all")
ft <- style(ft, ~ drat > 3.5, ~ vs + am + gear + carb,
  pr_t = fp_text(color="red", italic = TRUE) )

ft
```

 theme_alafoli

Apply alafoli theme

Description

Apply alafoli theme

Usage

```
theme_alafoli(x)
```

Arguments

x	a flextable object
---	--------------------

Illustrations

See Also

Other functions related to themes: [get_flextable_defaults\(\)](#), [set_flextable_defaults\(\)](#), [theme_booktabs\(\)](#), [theme_box\(\)](#), [theme_tron_legacy\(\)](#), [theme_tron\(\)](#), [theme_vader\(\)](#), [theme_vanilla\(\)](#), [theme_zebra\(\)](#)

Examples

```
ft <- flextable(head(airquality))
ft <- theme_alafoli(ft)
ft
```

theme_booktabs	<i>Apply booktabs theme</i>
----------------	-----------------------------

Description

Apply theme booktabs to a flextable

Usage

```
theme_booktabs(x, bold_header = FALSE, ...)
```

Arguments

x	a flextable object
bold_header	header will be bold if TRUE.
...	unused

Illustrations**See Also**

Other functions related to themes: [get_flextable_defaults\(\)](#), [set_flextable_defaults\(\)](#), [theme_alafoli\(\)](#), [theme_box\(\)](#), [theme_tron_legacy\(\)](#), [theme_tron\(\)](#), [theme_vader\(\)](#), [theme_vanilla\(\)](#), [theme_zebra\(\)](#)

Examples

```
ft <- flextable(head(airquality))
ft <- theme_booktabs(ft)
ft
```

theme_box	<i>Apply box theme</i>
-----------	------------------------

Description

Apply theme box to a flextable

Usage

```
theme_box(x)
```

Arguments

x a flextable object

Illustrations**See Also**

Other functions related to themes: [get_flextable_defaults\(\)](#), [set_flextable_defaults\(\)](#), [theme_alafoli\(\)](#), [theme_booktabs\(\)](#), [theme_tron_legacy\(\)](#), [theme_tron\(\)](#), [theme_vader\(\)](#), [theme_vanilla\(\)](#), [theme_zebra\(\)](#)

Examples

```
ft <- flextable(head(airquality))
ft <- theme_box(ft)
ft
```

theme_tron	<i>Apply tron theme</i>
------------	-------------------------

Description

Apply theme tron to a flextable

Usage

```
theme_tron(x)
```

Arguments

x a flextable object

Illustrations

See Also

Other functions related to themes: [get_flextable_defaults\(\)](#), [set_flextable_defaults\(\)](#), [theme_alafoli\(\)](#), [theme_booktabs\(\)](#), [theme_box\(\)](#), [theme_tron_legacy\(\)](#), [theme_vader\(\)](#), [theme_vanilla\(\)](#), [theme_zebra\(\)](#)

Examples

```
ft <- flextable(head(airquality))
ft <- theme_tron(ft)
ft
```

theme_tron_legacy	<i>Apply tron legacy theme</i>
-------------------	--------------------------------

Description

Apply theme tron legacy to a flextable

Usage

```
theme_tron_legacy(x)
```

Arguments

x a flextable object

Illustrations

See Also

Other functions related to themes: [get_flextable_defaults\(\)](#), [set_flextable_defaults\(\)](#), [theme_alafoli\(\)](#), [theme_booktabs\(\)](#), [theme_box\(\)](#), [theme_tron\(\)](#), [theme_vader\(\)](#), [theme_vanilla\(\)](#), [theme_zebra\(\)](#)

Examples

```
ft <- flextable(head(airquality))
ft <- theme_tron_legacy(ft)
ft
```

theme_vader	<i>Apply Sith Lord Darth Vader theme</i>
-------------	--

Description

Apply Sith Lord Darth Vader theme to a flextable

Usage

```
theme_vader(x, ...)
```

Arguments

x	a flextable object
...	unused

Illustrations**See Also**

Other functions related to themes: [get_flextable_defaults\(\)](#), [set_flextable_defaults\(\)](#), [theme_alafoli\(\)](#), [theme_booktabs\(\)](#), [theme_box\(\)](#), [theme_tron_legacy\(\)](#), [theme_tron\(\)](#), [theme_vanilla\(\)](#), [theme_zebra\(\)](#)

Examples

```
ft <- flextable(head(airquality))
ft <- theme_vader(ft)
ft
```

theme_vanilla	<i>Apply vanilla theme</i>
---------------	----------------------------

Description

Apply theme vanilla to a flextable: The external horizontal lines of the different parts of the table (body, header, footer) are black 2 points thick, the external horizontal lines of the different parts are black 0.5 point thick. Header text is bold, text columns are left aligned, other columns are right aligned.

Usage

```
theme_vanilla(x)
```

Arguments

x a flextable object

Illustrations**See Also**

Other functions related to themes: [get_flextable_defaults\(\)](#), [set_flextable_defaults\(\)](#), [theme_alafoli\(\)](#), [theme_booktabs\(\)](#), [theme_box\(\)](#), [theme_tron_legacy\(\)](#), [theme_tron\(\)](#), [theme_vader\(\)](#), [theme_zebra\(\)](#)

Examples

```
ft <- flextable(head(airquality))
ft <- theme_vanilla(ft)
ft
```

theme_zebra	<i>Apply zebra theme</i>
-------------	--------------------------

Description

Apply theme zebra to a flextable

Usage

```
theme_zebra(  
  x,  
  odd_header = "#CFCFCF",  
  odd_body = "#EFEFEF",  
  even_header = "transparent",  
  even_body = "transparent"  
)
```

Arguments

x a flextable object
odd_header, odd_body, even_header, even_body
odd/even colors for table header and body

Illustrations

See Also

Other functions related to themes: [get_flextable_defaults\(\)](#), [set_flextable_defaults\(\)](#), [theme_alafoli\(\)](#), [theme_booktabs\(\)](#), [theme_box\(\)](#), [theme_tron_legacy\(\)](#), [theme_tron\(\)](#), [theme_vader\(\)](#), [theme_vanilla\(\)](#)

Examples

```
ft <- flextable(head(airquality))
ft <- theme_zebra(ft)
ft
```

valign

Set vertical alignment

Description

change vertical alignment of selected rows and columns of a flextable.

Usage

```
valign(x, i = NULL, j = NULL, valign = "center", part = "body")
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
valign	vertical alignment of paragraph within cell, one of "center" or "top" or "bottom".
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations**See Also**

Other sugar functions for table style: [align\(\)](#), [bg\(\)](#), [bold\(\)](#), [color\(\)](#), [empty_blanks\(\)](#), [fontsize\(\)](#), [font\(\)](#), [highlight\(\)](#), [italic\(\)](#), [line_spacing\(\)](#), [padding\(\)](#), [rotate\(\)](#)

Examples

```
ft_1 <- flextable(iris[c(1:3, 51:53, 101:103),])
ft_1 <- theme_box(ft_1)
ft_1 <- merge_v(ft_1, j = 5)
ft_1

ft_2 <- valign(ft_1, j = 5, valign = "top", part = "all")
ft_2
```

vline	<i>set vertical borders</i>
-------	-----------------------------

Description

The function is applying vertical borders to inner content of one or all parts of a flextable. The lines are the right borders of selected cells.

Usage

```
vline(x, i = NULL, j = NULL, border = NULL, part = "all")
```

Arguments

x	a flextable object
i	rows selection
j	columns selection
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner_v\(\)](#), [border_inner\(\)](#), [border_outer\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline_top\(\)](#), [hline\(\)](#), [vline_left\(\)](#), [vline_right\(\)](#)

Examples

```
library(officer)
std_border = fp_border(color="orange")

ft <- flextable(head(iris))
ft <- border_remove(x = ft)

# add vertical borders
ft <- vline(ft, border = std_border )
ft
```

vline_left	<i>set flextable left vertical borders</i>
------------	--

Description

The function is applying vertical borders to the left side of one or all parts of a flextable. The line is the left border of selected cells of the first column.

Usage

```
vline_left(x, i = NULL, border = NULL, part = "all")
```

Arguments

x	a flextable object
i	rows selection
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner_v\(\)](#), [border_inner\(\)](#), [border_outer\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline_top\(\)](#), [hline\(\)](#), [vline_right\(\)](#), [vline\(\)](#)

Examples

```
library(officer)
std_border = fp_border(color="orange")

ft <- flextable(head(iris))
ft <- border_remove(x = ft)

# add vertical border on the left side of the table
ft <- vline_left(ft, border = std_border )
ft
```

vline_right	<i>set flextable right vertical borders</i>
-------------	---

Description

The function is applying vertical borders to the right side of one or all parts of a flextable. The line is the right border of selected cells of the last column.

Usage

```
vline_right(x, i = NULL, border = NULL, part = "all")
```

Arguments

x	a flextable object
i	rows selection
border	border properties defined by a call to fp_border()
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Illustrations

See Also

Other borders management: [border_inner_h\(\)](#), [border_inner_v\(\)](#), [border_inner\(\)](#), [border_outer\(\)](#), [border_remove\(\)](#), [border\(\)](#), [hline_bottom\(\)](#), [hline_top\(\)](#), [hline\(\)](#), [vline_left\(\)](#), [vline\(\)](#)

Examples

```
library(officer)
std_border = fp_border(color="orange")

ft <- flextable(head(iris))
ft <- border_remove(x = ft)

# add vertical border on the left side of the table
ft <- vline_right(ft, border = std_border )
ft
```

void	<i>Delete flextable content</i>
------	---------------------------------

Description

Set content display as a blank " " .

Usage

```
void(x, j = NULL, part = "body")
```

Arguments

x	flextable object
j	columns selection
part	partname of the table

Examples

```
ftab <- flextable(head(mtcars))
ftab <- void(ftab, ~ vs + am + gear + carb )
ftab
```

width	<i>Set flextable columns width</i>
-------	------------------------------------

Description

control columns width

Usage

```
width(x, j = NULL, width)
```

Arguments

x	flextable object
j	columns selection
width	width in inches

Details

Heights are not used when flextable is been rendered into HTML.

Illustrations

See Also

Other flextable dimensions: [autofit\(\)](#), [dim.flextable\(\)](#), [dim_pretty\(\)](#), [fit_to_width\(\)](#), [flextable_dim\(\)](#), [height\(\)](#), [hrule\(\)](#), [set_table_properties\(\)](#)

Examples

```
ft <- flextable(head(iris))
ft <- width(ft, width = 1.5)
ft
```

Index

- * **as_flextable methods**
 - as_flextable, 13
 - as_flextable.gam, 13
 - as_flextable.glm, 14
 - as_flextable.grouped_data, 15
 - as_flextable.htest, 16
 - as_flextable.lm, 17
 - as_flextable.xtable, 17
- * **borders management**
 - border_inner, 30
 - border_inner_h, 31
 - border_inner_v, 32
 - border_outer, 33
 - border_remove, 34
 - hline, 64
 - hline_bottom, 65
 - hline_top, 66
 - vline, 109
 - vline_left, 110
 - vline_right, 111
- * **cells formatters**
 - colformat_char, 34
 - colformat_date, 35
 - colformat_datetime, 36
 - colformat_double, 37
 - colformat_image, 38
 - colformat_int, 39
 - colformat_lgl, 40
 - colformat_num, 41
 - compose, 44
 - set_formatter, 97
- * **chunk elements for paragraph**
 - as_b, 10
 - as_bracket, 11
 - as_chunk, 12
 - as_i, 20
 - as_image, 21
 - as_sub, 24
 - as_sup, 25
 - gg_chunk, 60
 - hyperlink_text, 68
 - linrange, 73
 - lollipop, 75
 - minibar, 81
 - plot_chunk, 86
- * **flextable dimensions**
 - autofit, 26
 - dim.flextable, 46
 - dim_pretty, 47
 - fit_to_width, 49
 - flextable_dim, 52
 - height, 62
 - hrule, 67
 - set_table_properties, 100
 - width, 112
- * **flextable merging function**
 - merge_at, 77
 - merge_h, 78
 - merge_h_range, 78
 - merge_none, 79
 - merge_v, 80
- * **flextable print function**
 - as_raster, 23
 - flextable_to_rmd, 53
 - htmltools_value, 68
 - knit_print.flextable, 70
 - plot.flextable, 85
 - print.flextable, 87
 - save_as_docx, 90
 - save_as_html, 91
 - save_as_image, 92
 - save_as_pptx, 93
- * **functions for defining formatting properties**
 - fp_border_default, 58
 - fp_text_default, 58
- * **functions related to themes**
 - get_flextable_defaults, 59

- set_flextable_defaults, 95
- theme_alafoli, 102
- theme_booktabs, 103
- theme_box, 104
- theme_tron, 104
- theme_tron_legacy, 105
- theme_vader, 106
- theme_vanilla, 106
- theme_zebra, 107
- * headers and footers**
 - add_header, 6
 - add_header_lines, 7
 - add_header_row, 8
 - set_header_footer_df, 98
 - set_header_labels, 99
- * sugar functions for table style**
 - align, 9
 - bg, 27
 - bold, 29
 - color, 42
 - empty_blanks, 48
 - font, 54
 - fontsize, 55
 - highlight, 63
 - italic, 69
 - line_spacing, 75
 - padding, 82
 - rotate, 89
 - valign, 108
- add_body, 5
- add_footer (add_header), 6
- add_footer(), 5
- add_footer_lines (add_header_lines), 7
- add_footer_row (add_header_row), 8
- add_header, 6, 8, 9, 99, 100
- add_header(), 5
- add_header_lines, 6, 7, 9, 99, 100
- add_header_row, 6, 8, 8, 99, 100
- align, 9, 27, 29, 43, 48, 55, 63, 70, 75, 83, 89, 108
- align_nottext_col (align), 9
- align_text_col (align), 9
- as_b, 10, 11, 12, 21, 22, 24, 25, 60, 69, 74, 76, 82, 86
- as_bracket, 10, 11, 12, 21, 22, 24, 25, 60, 69, 74, 76, 82, 86
- as_chunk, 10, 11, 12, 21, 22, 24, 25, 60, 69, 74, 76, 82, 86
- as_chunk(), 23, 59
- as_flextable, 13, 13, 14–17, 19
- as_flextable.gam, 13, 13, 14–17, 19
- as_flextable.glm, 13, 14, 15–17, 19
- as_flextable.grouped_data, 13, 14, 15, 16, 17, 19
- as_flextable.grouped_data(), 20
- as_flextable.htest, 13–15, 16, 17, 19
- as_flextable.lm, 13–16, 17, 19
- as_flextable.xtable, 13–17, 17
- as_grouped_data, 19
- as_grouped_data(), 15
- as_i, 10–12, 20, 22, 24, 25, 60, 69, 74, 76, 82, 86
- as_image, 10–12, 21, 21, 24, 25, 60, 69, 74, 76, 82, 86
- as_image(), 23
- as_paragraph, 22
- as_paragraph(), 10–12, 21, 22, 24, 25, 44, 57, 60, 68, 73–76, 81, 82, 86
- as_raster, 23, 54, 68, 72, 85, 87, 90, 92, 93
- as_raster(), 85
- as_sub, 10–12, 21, 22, 24, 25, 60, 69, 74, 76, 82, 86
- as_sup, 10–12, 21, 22, 24, 25, 60, 69, 74, 76, 82, 86
- autofit, 26, 46, 47, 49, 52, 62, 67, 89, 101, 113
- autofit(), 51, 84
- bg, 10, 27, 29, 43, 48, 55, 63, 70, 75, 83, 89, 108
- body_add_flextable, 28
- body_replace_flextable_at_bkm (body_add_flextable), 28
- bold, 10, 27, 29, 43, 48, 55, 63, 70, 75, 83, 89, 108
- border, 30–34, 64–66, 109–111
- border_inner, 30, 31–34, 64–66, 109–111
- border_inner_h, 30, 31, 32–34, 64–66, 109–111
- border_inner_v, 30, 31, 32, 33, 34, 64–66, 109–111
- border_outer, 30–32, 33, 34, 64–66, 109–111
- border_remove, 30–33, 34, 64–66, 109–111
- colformat_char, 34, 36–42, 44, 98
- colformat_date, 35, 35, 37–42, 44, 98

- colformat_datetime, [35](#), [36](#), [36](#), [38–42](#), [44](#), [98](#)
- colformat_double, [35–37](#), [37](#), [39–42](#), [44](#), [98](#)
- colformat_double(), [98](#)
- colformat_image, [35–38](#), [38](#), [40–42](#), [44](#), [98](#)
- colformat_int, [35–39](#), [39](#), [41](#), [42](#), [44](#), [98](#)
- colformat_int(), [96](#)
- colformat_lgl, [35–40](#), [40](#), [42](#), [44](#), [98](#)
- colformat_num, [35–41](#), [41](#), [44](#), [98](#)
- colformat_num(), [5](#), [6](#), [96](#)
- color, [10](#), [27](#), [29](#), [42](#), [48](#), [55](#), [63](#), [70](#), [75](#), [83](#), [89](#), [108](#)
- compose, [35–42](#), [44](#), [98](#)
- compose(), [4](#), [12](#), [21](#), [22](#), [51](#), [60](#), [68](#), [69](#), [73–76](#), [81](#), [82](#), [86](#)
- continuous_summary, [45](#)
- delete_part, [46](#)
- dim.flextable, [26](#), [46](#), [47](#), [49](#), [52](#), [62](#), [67](#), [101](#), [113](#)
- dim_pretty, [26](#), [46](#), [47](#), [49](#), [52](#), [62](#), [67](#), [89](#), [101](#), [113](#)
- dim_pretty(), [26](#), [84](#)
- div(), [68](#)
- empty_blanks, [10](#), [27](#), [29](#), [43](#), [48](#), [55](#), [63](#), [70](#), [75](#), [83](#), [89](#), [108](#)
- fit_to_width, [26](#), [46](#), [47](#), [49](#), [52](#), [62](#), [67](#), [101](#), [113](#)
- fix_border_issues, [50](#)
- flextable, [50](#)
- flextable(), [4](#), [5](#), [94](#)
- flextable-package, [4](#)
- flextable_dim, [26](#), [46](#), [47](#), [49](#), [52](#), [62](#), [67](#), [101](#), [113](#)
- flextable_html_dependency, [52](#)
- flextable_to_rmd, [24](#), [53](#), [68](#), [72](#), [85](#), [87](#), [90](#), [92](#), [93](#)
- font, [10](#), [27](#), [29](#), [43](#), [48](#), [54](#), [55](#), [63](#), [70](#), [75](#), [83](#), [89](#), [108](#)
- fontsize, [10](#), [27](#), [29](#), [43](#), [48](#), [55](#), [55](#), [63](#), [70](#), [75](#), [83](#), [89](#), [108](#)
- footers_flextable_at_bkm, [56](#)
- footnote, [56](#)
- footnote(), [51](#)
- format(), [38](#), [40](#), [42](#)
- formatC, [96](#)
- fp_border(), [30–33](#), [58](#), [64–66](#), [109–111](#)
- fp_border_default, [58](#), [59](#)
- fp_text(), [58](#)
- fp_text_default, [58](#), [58](#)
- get_flextable_defaults, [59](#), [96](#), [103–108](#)
- get_flextable_defaults(), [95](#)
- gg_chunk, [10–12](#), [21](#), [22](#), [24](#), [25](#), [60](#), [69](#), [74](#), [76](#), [82](#), [86](#)
- headers_flextable_at_bkm, [61](#)
- height, [26](#), [46](#), [47](#), [49](#), [52](#), [62](#), [67](#), [101](#), [113](#)
- height(), [84](#)
- height_all(height), [62](#)
- highlight, [10](#), [27](#), [29](#), [43](#), [48](#), [55](#), [63](#), [70](#), [75](#), [83](#), [89](#), [108](#)
- hline, [30–34](#), [64](#), [65](#), [66](#), [109–111](#)
- hline(), [58](#)
- hline_bottom, [30–34](#), [64](#), [65](#), [66](#), [109–111](#)
- hline_top, [30–34](#), [64](#), [65](#), [66](#), [109–111](#)
- hrule, [26](#), [46](#), [47](#), [49](#), [52](#), [62](#), [67](#), [101](#), [113](#)
- hrule(), [62](#)
- HTML, [68](#)
- htmltools_value, [24](#), [54](#), [68](#), [72](#), [85](#), [87](#), [90](#), [92](#), [93](#)
- hyperlink_text, [10–12](#), [21](#), [22](#), [24](#), [25](#), [60](#), [68](#), [74](#), [76](#), [82](#), [86](#)
- hyperlink_text(), [23](#)
- italic, [10](#), [27](#), [29](#), [43](#), [48](#), [55](#), [63](#), [69](#), [75](#), [83](#), [89](#), [108](#)
- knit_print.flextable, [24](#), [53](#), [54](#), [68](#), [70](#), [85](#), [87](#), [90](#), [92](#), [93](#)
- knit_print.flextable(), [51](#), [87](#)
- line_spacing, [10](#), [27](#), [29](#), [43](#), [48](#), [55](#), [63](#), [70](#), [75](#), [83](#), [89](#), [108](#)
- linerrange, [10–12](#), [21](#), [22](#), [24](#), [25](#), [60](#), [69](#), [73](#), [76](#), [82](#), [86](#)
- lollipop, [10–12](#), [21](#), [22](#), [24](#), [25](#), [60](#), [69](#), [74](#), [75](#), [82](#), [86](#)
- merge_at, [77](#), [78–80](#)
- merge_h, [77](#), [78](#), [79](#), [80](#)
- merge_h(), [6](#)
- merge_h_range, [77](#), [78](#), [78](#), [79](#), [80](#)
- merge_none, [77–79](#), [79](#), [80](#)
- merge_v, [77–79](#), [80](#)
- merge_v(), [6](#)

- minibar, [10–12](#), [21](#), [22](#), [24](#), [25](#), [60](#), [69](#), [74](#), [76](#), [81](#), [86](#)
- minibar(), [23](#)
- mk_par (compose), [44](#)
- officer::fp_text(), [12](#), [69](#)
- officer::ph_location_type(), [84](#)
- officer::read_pptx(), [84](#)
- officer::run_autonum(), [94](#)
- officer::styles_info(), [94](#)
- padding, [10](#), [27](#), [29](#), [43](#), [48](#), [55](#), [63](#), [70](#), [75](#), [82](#), [89](#), [108](#)
- par(), [86](#)
- ph_with.flextable, [84](#)
- plot.flextable, [24](#), [54](#), [68](#), [72](#), [85](#), [87](#), [90](#), [92](#), [93](#)
- plot_chunk, [10–12](#), [21](#), [22](#), [24](#), [25](#), [60](#), [69](#), [74](#), [76](#), [82](#), [86](#)
- print.flextable, [24](#), [54](#), [68](#), [72](#), [85](#), [87](#), [90](#), [92](#), [93](#)
- proc_freq, [88](#)
- prop_section, [90](#)
- qflextable (flextable), [50](#)
- regulartable (flextable), [50](#)
- rotate, [10](#), [27](#), [29](#), [43](#), [48](#), [55](#), [63](#), [70](#), [75](#), [83](#), [89](#), [108](#)
- save_as_docx, [24](#), [54](#), [68](#), [72](#), [85](#), [87](#), [90](#), [92](#), [93](#)
- save_as_html, [24](#), [54](#), [68](#), [72](#), [85](#), [87](#), [90](#), [91](#), [93](#)
- save_as_image, [24](#), [54](#), [68](#), [72](#), [85](#), [87](#), [90](#), [92](#), [92](#), [93](#)
- save_as_pptx, [24](#), [54](#), [68](#), [72](#), [85](#), [87](#), [90](#), [92](#), [93](#), [93](#)
- set_caption, [94](#)
- set_caption(), [51](#)
- set_flextable_defaults, [60](#), [95](#), [103–108](#)
- set_flextable_defaults(), [50](#), [51](#), [58](#)
- set_footer_df (set_header_footer_df), [98](#)
- set_formatter, [35–42](#), [44](#), [97](#)
- set_formatter_type (set_formatter), [97](#)
- set_header_df (set_header_footer_df), [98](#)
- set_header_footer_df, [6](#), [8](#), [9](#), [98](#), [100](#)
- set_header_labels, [6](#), [8](#), [9](#), [99](#), [99](#)
- set_table_properties, [26](#), [46](#), [47](#), [49](#), [52](#), [62](#), [67](#), [100](#), [113](#)
- strptime(), [36](#), [37](#), [96](#)
- style, [101](#)
- style(), [51](#)
- theme_alafoli, [60](#), [96](#), [102](#), [103–108](#)
- theme_booktabs, [60](#), [96](#), [103](#), [103](#), [104–108](#)
- theme_booktabs(), [51](#)
- theme_box, [60](#), [96](#), [103](#), [104](#), [105–108](#)
- theme_tron, [60](#), [96](#), [103](#), [104](#), [104](#), [105–108](#)
- theme_tron_legacy, [60](#), [96](#), [103–105](#), [105](#), [106–108](#)
- theme_vader, [60](#), [96](#), [103–105](#), [106](#), [107](#), [108](#)
- theme_vanilla, [60](#), [96](#), [103–106](#), [106](#), [108](#)
- theme_zebra, [60](#), [96](#), [103–107](#), [107](#)
- valign, [10](#), [27](#), [29](#), [43](#), [48](#), [55](#), [63](#), [70](#), [75](#), [83](#), [89](#), [108](#)
- vline, [30–34](#), [64–66](#), [109](#), [110](#), [111](#)
- vline(), [58](#)
- vline_left, [30–34](#), [64–66](#), [109](#), [110](#), [111](#)
- vline_right, [30–34](#), [64–66](#), [109](#), [110](#), [111](#)
- void, [112](#)
- width, [26](#), [46](#), [47](#), [49](#), [52](#), [62](#), [67](#), [89](#), [101](#), [112](#)
- width(), [84](#)
- xtable_to_flextable
(as_flextable.xtable), [17](#)