

Package ‘flextable’

October 31, 2018

Type Package

Title Functions for Tabular Reporting

Version 0.4.6

Description Create pretty tables for 'HTML', 'Microsoft Word' and 'Microsoft PowerPoint' documents. Functions are provided to let users create tables, modify and format their content. It extends package 'officer' that does not contain any feature for customized tabular reporting and can be used within R markdown documents when rendering to HTML and to 'Microsoft Word' documents.

License GPL-3

LazyData TRUE

Imports stats,utils,grDevices, officer (>= 0.2.0), rmarkdown, knitr,htmltools, R6,stringr,xml2, gdtools (>= 0.1.6)

RoxygenNote 6.1.0

Suggests testthat, xtable

Encoding UTF-8

URL <https://davidgohel.github.io/flextable>

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

VignetteBuilder knitr

NeedsCompilation no

Author David Gohel [aut, cre],
Maxim Nazarov [ctb] (rmarkdown for docx output)

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

Repository CRAN

Date/Publication 2018-10-31 16:40:03 UTC

R topics documented:

flextable-package	3
add_header_footer	3

align	4
as_image	5
autofit	5
bg	6
body_add_flextable	7
bold	8
border	8
borders	9
colformat_char	12
colformat_int	13
colformat_num	13
color	14
delete_part	15
dim.flextable	16
dim_pretty	16
display	17
empty_blanks	18
flextable	19
font	20
fontsize	20
footers_flextable_at_bkm	21
format.flextable	22
headers_flextable_at_bkm	22
height	23
htmltools_value	23
hyperlink_text	24
italic	25
merge_at	26
merge_h	26
merge_none	27
merge_v	27
minibar	28
padding	28
ph_with_flextable	29
print.flextable	30
rotate	31
set_formatter	32
set_header_footer_df	33
set_header_labels	34
style	34
theme_booktabs	35
theme_box	35
theme_tron	36
theme_tron_legacy	36
theme_vanilla	37
theme_zebra	37
void	38
width	38

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 `display` lets customise text of cells. `regulartable()` function has been written because the first one is resource consuming. The main difference is that it is producing tables where cells can contain only one chunk of text with its own set of formatting properties. Function `set_formatter` is customizing text of cells. `regulartable` is to be used in most of cases. These two type of table DO NOT ACCEPT new lines in text.

See Also

<https://davidgohe1.github.io/flextable/>, `regulartable`, `flextable`

Description

Add a single row of labels in the flextable’s header or footer part. It can be inserted at the top or the bottom of the part.

Usage

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

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

Arguments

- `x` a flextable object
- `top` should the row be inserted at the top or the bottom.
- `...` a named list (names are data colnames) of strings specifying corresponding labels to add.

Examples

```
ft <- flextable( head( iris ),
  col_keys = c("Species", "Sepal.Length", "Petal.Length", "Sepal.Width", "Petal.Width") )
ft <- add_header(x = ft, Sepal.Length = "length",
  Sepal.Width = "width", Petal.Length = "length",
  Petal.Width = "width", Species = "Species", top = FALSE )
ft <- add_header(ft, Sepal.Length = "Inches",
  Sepal.Width = "Inches", Petal.Length = "Inches",
  Petal.Width = "Inches", Species = "Species", top = TRUE )
ft <- merge_h(ft, part = "header")
ft <- add_footer(ft, Species = "This is a footnote" )
ft <- merge_at(ft, j = 1:5, part = "footer")
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")
```

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')

Examples

```
ft <- flextable(mtcars)
ft <- align(ft, align = "center")
```

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

Description

The function has to be used with function `display()`. It lets add images within flextable.

Usage

```
as_image(x, src, width = 1, height = 0.2)
```

Arguments

x	image will be repeated while iterating this variable. The variable should be one of the original data frame. Note its values are not used, only its size.
src	image filename
width, height	size of the png file in inches

See Also

[display](#)

Examples

```
library(officer)
img.file <- file.path( R.home("doc"), "html", "logo.jpg" )
data <- head( mtcars, n = 10)
data$img_src <- rep(img.file, 10)
myft <- flextable(data)
myft <- display(myft,
  i = ~ qsec > 18, col_key = "qsec", pattern = "{{r_logo}}",
  formatters = list( r_logo ~ as_image(qsec,
    src = img_src, width = .20, height = .15)),
  fprops = list(qsec = fp_text(color = "orange")) )
myft
```

autofit	<i>Adjusts cell widths and heights</i>
---------	--

Description

compute and apply optimized widths and heights.

Usage

```
autofit(x, add_w = 0.1, add_h = 0.1)
```

Arguments

x	flextable object
add_w	extra width to add in inches
add_h	extra height to add in inches

Examples

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

bg	<i>Set background color</i>
----	-----------------------------

Description

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

Usage

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

Arguments

x	a flextable object
i	rows selection
j	columns selection
bg	color to use as background color
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Examples

```
ft <- flextable(mtcars)
ft <- bg(ft, bg = "#DDDDDD", part = "header")
```

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)
ft <- flextable(head(mtcars))
ft <- theme_zebra(ft)
ft <- autofit(ft)
doc <- read_docx()
doc <- body_add_flextable(doc, value = ft)
print(doc, target = "test.docx")
```

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

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')

Examples

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

border	<i>Set cell borders</i>
--------	-------------------------

Description

change borders of selected rows and columns of a flextable.

Usage

```
border(x, i = NULL, j = NULL, border = NULL, border.top = NULL,
  border.bottom = NULL, border.left = NULL, border.right = NULL,
  part = "body")
```


Arguments

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

Note

This function should not be used directly by users and functions [hline](#), [vline](#), [hline_top](#), [vline_left](#) should be preferred.

Examples

```
library(officer)
ft <- flextable(mtcars)
ft <- border(ft, border.top = fp_border(color = "orange") )
```

borders	<i>borders management</i>
---------	---------------------------

Description

These functions let control the horizontal or vertical borders of a flextable. They are sugar functions and should be used instead of function [border](#) that requires careful settings to avoid overlapping borders.

Usage

```
border_remove(x)

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

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

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

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

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

```
hline_bottom(x, j = NULL, border = NULL, part = "body")
vline(x, i = NULL, j = NULL, border = NULL, part = "all")
vline_left(x, i = NULL, border = NULL, part = "all")
vline_right(x, i = NULL, border = NULL, part = "all")
```

Arguments

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

border_remove

The function is deleting all borders of the flextable object.

border_outer

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

border_inner_h

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

border_inner_v

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

hline

The function is setting horizontal lines along the part part of the flextable object. The lines are the bottom borders of selected cells.

hline_top

The function is setting the first horizontal line of the part part of the flextable object. The line is the top border of selected cells of the first row.

hline_bottom

The function is setting the last horizontal line of the part part of the flextable object. The line is the bottom border of selected cells of the last row.

vline

The function is setting vertical lines along the part part of the flextable object. The lines are the right borders of selected cells.

vline_left

The function is setting the first vertical line of the part part of the flextable object. The line is the left border of selected cells of the first column.

vline_right

The function is setting the last vertical line of the part part of the flextable object. The line is the right border of selected cells of the last column.

Examples

```
# need officer to define border properties
library(officer)
big_border = fp_border(color="red", width = 2)
std_border = fp_border(color="orange", width = 1)

# dataset to be used for examples
dat <- iris[c(1:5, 51:55, 101:105),]

# use of regulartable() to create a table
ft <- regulartable(dat)

# remove all borders
ft <- border_remove(x = ft)

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

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

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

# new example
ft <- regulartable(dat, col_keys = c("Species", "Sepal.Length",
  "Sepal.Width", "Petal.Length", "Petal.Width" ))
ft <- border_remove(x = ft)

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

```

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

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

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

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

# add vertical border on the right side of the table
ft <- vline_right(ft, border = big_border )
ft

```

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

Description

Format character columns in a flextable or regulartable.

Usage

```
colformat_char(x, col_keys, na_str = "", prefix = "", suffix = "")
```

Arguments

x	a regulartable object
col_keys	names of the colkeys
na_str	string to be used for NA values
prefix, suffix	string to be used as prefix or suffix

See Also

Other columns formatters: [colformat_int](#), [colformat_num](#)

Examples

```

dat <- iris
ft <- regulartable(dat)
ft <- colformat_char(
  x = ft, col_keys = "Species", suffix = "!!")
autofit(ft)

```

colformat_int	<i>format integer columns</i>
---------------	-------------------------------

Description

Format integer columns in a flextable or regulartable.

Usage

```
colformat_int(x, col_keys, big.mark = ",", na_str = "", prefix = "",  
             suffix = "")
```

Arguments

x	a regulartable object
col_keys	names of the colkeys
big.mark	see formatC
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 columns formatters: [colformat_char](#), [colformat_num](#)

Examples

```
dat <- mtcars  
  
ft <- regulartable(dat)  
colkeys <- c("vs", "am", "gear", "carb")  
ft <- colformat_int(x = ft, col_keys = colkeys, prefix = "# ")  
autofit(ft)
```

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

Description

Format numeric columns in a flextable or regulartable.

Usage

```
colformat_num(x, col_keys, big.mark = ",", digits = 2, na_str = "",  
             prefix = "", suffix = "")
```

Arguments

x a regular table object
col_keys names of the colkeys
big.mark, digits see [formatC](#)
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 columns formatters: [colformat_char](#), [colformat_int](#)

Examples

```

dat <- iris
dat[1:4, 1] <- NA
dat[, 2] <- dat[, 2] * 1000000

ft <- regularTable(dat)
colkeys = c("Sepal.Length", "Sepal.Width",
            "Petal.Length", "Petal.Width")
ft <- colformat_num(
  x = ft, col_keys = colkeys,
  big.mark="," , digits = 2, na_str = "N/A")
autofit(ft)

ft <- flexTable(dat)
colkeys = c("Sepal.Length", "Sepal.Width",
            "Petal.Length", "Petal.Width")
ft <- colformat_num(
  x = ft, col_keys = colkeys,
  big.mark="," , digits = 2, na_str = "N/A")
autofit(ft)

```

color

Set font color

Description

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

Usage

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

Arguments

x	a flextable object
i	rows selection
j	columns selection
color	color to use as font color
part	partname of the table (one of 'all', 'body', 'header', 'footer')

Examples

```
ft <- flextable(mtcars)
ft <- color(ft, color = "orange", part = "header")
```

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').

Examples

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

dim.flextable *Get flextable dimensions*

Description

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

Usage

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

Arguments

x flextable object

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')

Examples

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

display	<i>Define flextable displayed values</i>
---------	--

Description

Modify flextable displayed values by specifying a string expression. Function is handling complex formatting as well as image insertion.

Usage

```
display(x, i = NULL, col_key, pattern, formatters = list(),
        fprops = list(), part = "body")
```

Arguments

x	a flextable object
i	rows selection
col_key	column to modify, a single character
pattern	string to format
formatters	a list of formula, left side for the name, right side for the content.
fprops	a named list of fp_text
part	partname of the table (one of 'all', 'body', 'header', 'footer')

pattern

It defined the template used to format the produced strings. Names enclosed by double braces will be evaluated as R code, the corresponding R code is defined with the argument `formatters`.

formatters

Each compound is specifying the R code to execute to produce strings that will be substituted in the `pattern` argument. An element must be a formula: the left-hand side is a name (matching a name enclosed by double braces in `pattern`) and the right-hand side is an R expression to be evaluated (that will produce the corresponding strings).

The function is designed to work with columns in the dataset provided to `flextable` (the `col_keys`).

fprops

A named list of [fp_text](#). It defines the formatting properties associated to a compound in `formatters`. If not defined for an element of `formatters`, the default formatting properties will be applied.

Note

The function `display` only works with `flextable` objects, use [set_formatter](#) for `regulartable` objects.

Examples

```
library(officer)
# Formatting data values example -----
ft <- flextable(head( mtcars, n = 10))
ft <- display(ft, col_key = "carb",
  i = ~ drat > 3.5, pattern = "# {{carb}}",
  formatters = list(carb ~ sprintf("%.1f", carb)),
  fprops = list(carb = fp_text(color="orange") ) )
ft <- autofit(ft)
```

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

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

ft <- flextable(head(iris), col_keys = c("Species",
  "break1", "Sepal.Length", "Sepal.Width",
  "break2", "Petal.Length", "Petal.Width" ) )
ft <- set_header_df(ft, mapping = typology, key = "col_keys" )
ft <- merge_h(ft, part = "header")
ft <- theme_vanilla(ft)
ft <- empty_blanks(ft)
ft <- width(ft, j = c(2, 5), width = .1 )
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 easily be defined, rows heights and columns widths can be manually set or automatically computed.

Usage

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

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

Arguments

<code>data</code>	dataset
<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>cwidth, cheight</code>	initial width and height to use for cell sizes in inches.

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.

Examples

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

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')

Examples

```
require("gdtools")
fontname <- "Times"

if( !font_family_exists(fontname) ){
  # if Times is not available, we will use the first available
  font_list <- sys_fonts()
  fontname <- as.character(font_list$family[1])
}

ft <- regulartable(head(iris))
ft <- font(ft, fontname = fontname, part = "header")
```

fontsize	<i>Set font size</i>
----------	----------------------

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')

Examples

```
ft <- flextable(mtcars)
ft <- fontsize(ft, size = 14, part = "header")
```

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

```
format.flextable      Encode flextable in a document format.
```

Description

Encode flextable in a document format, html, docx, pptx.

This function is exported so that users can create their own custom component.

Usage

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

Arguments

x	flextable object
type	one of pptx, docx or html.
...	unused

Examples

```
ft <- flextable(head(iris, n = 2))
format(ft, type = "html")
```

```
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.

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

Examples

```
ft <- flextable(iris)
ft <- height(ft, height = .3)
```

htmltools_value	<i>Render flextable in rmarkdown (including Word output)</i>
-----------------	--

Description

Function used to render flextable in knitr/rmarkdown documents. HTML and Word outputs are supported.

Function `htmltools_value` return an HTML version of the flextable, this function is to be used within Shiny applications with `renderUI()`.

Usage

```
htmltools_value(x)
```

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

Arguments

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

Note

For Word (docx) output, if pandoc version ≥ 2.0 is used, a raw XML block with the table code will be inserted. If pandoc version < 2.0 is used, an error will be raised. Note also that insertion of images is not supported with rmarkdown for Word documents.

Result can be aligned with chunk option `ft.align` that accepts values 'left', 'center' and 'right'. Word option 'Allow row to break across pages' can be activated with chunk option `ft.split` set to TRUE.

Author(s)

Maxim Nazarov

hyperlink_text	<i>hyperlink wrapper</i>
----------------	--------------------------

Description

The function has to be used with function `display()`. It lets add hyperlinks within flextable.

Usage

```
hyperlink_text(url, label = url)
```

Arguments

url, label url and label to be used

See Also

[display](#)

Examples

```
dat <- data.frame(
  col = "CRAN website", href = "https://cran.r-project.org",
  stringsAsFactors = FALSE)

ft <- flextable(dat)
ft <- display(
  ft, col_key = "col", pattern = "# {{mylink}}",
  formatters = list(mylink ~ hyperlink_text(href, col) )
)
ft
```



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

ft <- flextable(dat)
ft <- display( x = ft,
  col_key = "col",
  pattern = "This is a link: {{mylink}}",
  formatters = list(mylink ~ hyperlink_text(href, col) ) )
ft
```

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')

Examples

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

merge_at	<i>Merge flextable cells</i>
----------	------------------------------

Description

Merge flextable cells

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.

Examples

```
ft_merge <- flextable( head( mtcars ), cwidth = .5 )
ft_merge <- merge_at( ft_merge, i = 1:2, j = 1:3 )
ft_merge
```

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

Description

Merge flextable cells horizontally when consecutive cells have identical 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.

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_none	<i>Delete flextable merging informations</i>
------------	--

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.

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	<i>Merge flextable cells vertically</i>
---------	---

Description

Merge flextable cells vertically when consecutive cells have identical values.

Usage

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

Arguments

x	flextable object
j	columns names/keys where cells have to be merged.
part	partname of the table where merge has to be done.

Examples

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

minibar	<i>draw a single bar</i>
---------	--------------------------

Description

This function is used to insert bars into flextable with function display

Usage

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

Arguments

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

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')

Examples

```
ft <- flextable(mtcars)
ft <- padding(ft, padding.top = 4)
```

ph_with_flextable *add flextable into a PowerPoint slide*

Description

add a flextable as a new shape in the current slide.

Usage

```
ph_with_flextable(x, value, type = "body", index = 1)

ph_with_flextable_at(x, value, left, top)
```

Arguments

x	an rpptx device
value	flextable object
type	placeholder type
index	placeholder index (integer). This is to be used when a placeholder type is not unique in the current slide, e.g. two placeholders with type 'body'.
left, top	location of flextable on the slide in inches

Note

The width and height of the table can not be set with this function. 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(mtcars))

doc <- read_pptx()
doc <- add_slide(doc, layout = "Title and Content",
                master = "Office Theme")
doc <- ph_with_flextable(doc, value = ft, type = "body")
doc <- ph_with_flextable_at(doc, value = ft, left = 4, top = 5)
print(doc, target = "test.pptx" )

```

print.flextable	<i>flextable printing</i>
-----------------	---------------------------

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](#).

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

Description

apply a rotation to cell text

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", "tbl", "blr"
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

One common case is to rotate text to minimise column space. When rotating, paragraph alignments will remain the same and often right aligned (with an effect of top aligned when rotated). Use `align(..., align = "center")` to center rotated text.

When function `autofit` is used, the rotation will be ignored.

Examples

```
ft <- flextable(head(iris))
ft <- rotate(ft, rotation = "tbl", part = "header", align = "center")
ft <- align(ft, align = "center")
ft <- autofit(ft)
ft <- height(ft, height = max(dim_pretty(ft, part = "header")$widths), part = "header")
```

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, ..., 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",
  na_str = "")
```

Arguments

`x` a regulartable object

`...` Name-value pairs of functions, names should be existing `col_key` values

`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.

`na_str` string for NA values

set_formatter_type

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

Examples

```
ft <- regulartable( 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.

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_h(ft, part = "header")
ft <- merge_v(ft, j = "Species", part = "header")
ft <- theme_vanilla(ft)
```

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

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, ...)
```

Arguments

`x` a flextable object
`...` a named list (names are colnames), each element is a single character value specifying label to use.

Examples

```
ft_1 <- flextable( head( iris ))
ft_1 <- set_header_labels(ft_1, Sepal.Length = "Sepal length",
  Sepal.Width = "Sepal width", Petal.Length = "Petal length",
  Petal.Width = "Petal width"
)
ft_1
```

style *Set flextable style*

Description

Modify flextable text, paragraphs and cells formatting properties.

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')

Examples

```
library(officer)
def_cell <- fp_cell(border = fp_border(color="#00FFFF"))

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

ft <- flextable(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_booktabs	<i>Apply booktabs theme</i>
----------------	-----------------------------

Description

Apply theme tron to a flextable

Usage

```
theme_booktabs(x)
```

Arguments

x a flextable object

Examples

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

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

Description

Apply theme box to a flextable

Usage

```
theme_box(x)
```

Arguments

x a flextable object

Examples

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

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

Description

Apply theme tron to a flextable

Usage

```
theme_tron(x)
```

Arguments

x a flextable object

Examples

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

Examples

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

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

Description

Apply theme vanilla to a flextable

Usage

```
theme_vanilla(x)
```

Arguments

x a flextable object

Examples

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

Examples

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

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

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.

See Also

[flextable](#)

Examples

```
ft <- flexible(iris)
ft <- width(ft, width = 1)
```

xtable_to_flexible *get a flexible from a xtable object*

Description

Get a flexible object from a xtable object.

Usage

```
xtable_to_flexible(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.

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|clr|r|"
ft <- xtable_to_flexitable(
  tli.table,
  rotate.colnames = TRUE,
  include.rownames = FALSE)
ft <- height(ft, i = 1, part = "header", height = 1)
ft

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

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

```


Index

`add_footer` (`add_header_footer`), 3
`add_header` (`add_header_footer`), 3
`add_header_footer`, 3
`align`, 4
`as_image`, 5
`autofit`, 5, 29

`bg`, 6
`body_add_flextable`, 7
`body_replace_flextable_at_bkm`
 (`body_add_flextable`), 7
`bold`, 8
`border`, 8, 9
`border_inner_h` (`borders`), 9
`border_inner_v` (`borders`), 9
`border_outer` (`borders`), 9
`border_remove` (`borders`), 9
`borders`, 9

`colformat_char`, 12, 13, 14
`colformat_int`, 12, 13, 14
`colformat_num`, 12, 13, 13
`color`, 14

`delete_part`, 15
`dim.flextable`, 16
`dim_pretty`, 16, 29
`display`, 3, 5, 17, 24

`empty_blanks`, 18

`flextable`, 3, 19, 38
`flextable-package`, 3
`font`, 20
`fontsize`, 20
`footers` (`set_header_footer_df`), 33
`footers_flextable_at_bkm`, 21
`format.flextable`, 22
`formatC`, 13, 14
`fp_text`, 17

`headers` (`set_header_footer_df`), 33
`headers_flextable_at_bkm`, 22
`height`, 23, 29
`height_all` (`height`), 23
`hline`, 9
`hline` (`borders`), 9
`hline_bottom` (`borders`), 9
`hline_top`, 9
`hline_top` (`borders`), 9
`htmltools_value`, 23
`hyperlink_text`, 24

`italic`, 25

`knit_print.flextable`, 30
`knit_print.flextable` (`htmltools_value`),
 23

`merge_at`, 26
`merge_h`, 26
`merge_none`, 27
`merge_v`, 27
`minibar`, 28

`padding`, 28
`ph_with_flextable`, 29
`ph_with_flextable_at`
 (`ph_with_flextable`), 29
`print.flextable`, 30

`regulartable`, 3
`regulartable` (`flextable`), 19
`rotate`, 31

`set_footer_df` (`set_header_footer_df`), 33
`set_formatter`, 3, 17, 32
`set_formatter_type` (`set_formatter`), 32
`set_header_df` (`set_header_footer_df`), 33
`set_header_footer_df`, 33
`set_header_labels`, 34
`style`, 34

theme_booktabs, [35](#)
theme_box, [35](#)
theme_tron, [36](#)
theme_tron_legacy, [36](#)
theme_vanilla, [37](#)
theme_zebra, [37](#)

vline, [9](#)
vline(borders), [9](#)
vline_left, [9](#)
vline_left(borders), [9](#)
vline_right(borders), [9](#)
void, [38](#)

width, [29](#), [38](#)

xtable_to_flexable, [39](#)