

# Package ‘flextable’

February 8, 2019

**Type** Package

**Title** Functions for Tabular Reporting

**Version** 0.5.1

**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.

**License** GPL-3

**LazyData** TRUE

**Imports** stats, utils, grDevices, officer (>= 0.2.0), rmarkdown, knitr, htmltools, xml2, data.table, gdtools (>= 0.1.6), rlang, magick

**RoxygenNote** 6.1.1

**Suggests** testthat, xtable, magrittr

**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** 2019-02-07 23:13:44 UTC

## R topics documented:

flextable-package . . . . .	3
add_header_footer . . . . .	4
align . . . . .	5

as_chunk	6
as_flextable	7
as_grouped_data	8
as_image	8
as_paragraph	9
autofit	10
bg	11
body_add_flextable	11
bold	12
border	13
border_inner_h	14
border_inner_v	14
border_outer	15
border_remove	16
colformat_char	17
colformat_int	17
colformat_lgl	18
colformat_num	19
color	20
compose	21
delete_part	21
dim.flextable	22
dim_pretty	22
display	23
empty_blanks	24
fix_border_issues	25
flextable	25
font	26
fontsize	27
footers_flextable_at_bkm	28
format.flextable	28
headers_flextable_at_bkm	29
height	29
hline	30
hline_bottom	31
hline_top	31
htmltools_value	32
hyperlink_text	33
italic	34
merge_at	35
merge_h	35
merge_h_range	36
merge_none	37
merge_v	37
minibar	38
padding	39
ph_with_flextable	40
print.flextable	41

rotate . . . . .	41
set_caption . . . . .	42
set_formatter . . . . .	43
set_header_footer_df . . . . .	44
set_header_labels . . . . .	45
style . . . . .	45
theme_alafoli . . . . .	46
theme_booktabs . . . . .	47
theme_box . . . . .	47
theme_tron . . . . .	48
theme_tron_legacy . . . . .	48
theme_vader . . . . .	49
theme_vanilla . . . . .	50
theme_zebra . . . . .	50
vline . . . . .	51
vline_left . . . . .	52
vline_right . . . . .	52
void . . . . .	53
width . . . . .	54
xtable_to_flextable . . . . .	54

**Index****57**


---

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\_header\_footer      *Add a row of labels in header or footer part*

---

### 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_header_row(x, top = TRUE, values = character(),
  colwidths = integer())

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

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

add_header_lines(x, values = character(), 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.
values	values to add as a character vector
colwidths	the number of columns to merge in the row for each label

### 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
```

```

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

```
align_nottext_col(x, align = "right", header = 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

**Examples**

```
ft <- flextable(mtcars)
ft <- align(ft, align = "center")
ft <- flextable(mtcars)
ft <- align_text_col(ft, align = "left")
ft <- align_nottext_col(ft, align = "right")
ft
```

---

as\_chunk

*chunk of text wrapper*


---

**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, formater = format_fun, ...)
```

**Arguments**

x	text or any element that can be formatted as text with function provided in argument formater.
props	an <a href="#">fp_text</a> object to be used to format the text. If not specified, it will be the default value corresponding to the cell.
formater	a function that will format x as a character vector.
...	additional arguments for formater function.

**See Also**

Other chunk elements for paragraph: [as\\_image](#), [hyperlink\\_text](#), [minibar](#)

**Examples**

```
library(officer)

myft <- flextable( head(iris))

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

---

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

## Arguments

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

## Examples

```
# as_flextable and as_grouped_data -----  
if( require("magrittr")){  
  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 <- as_grouped_data(x = data_co2, groups = c("Treatment"))  
  
  zz <- as_flextable( data_co2 ) %>%  
    add_footer_lines("dataset C02 has been used for this flextable") %>%  
    add_header_lines("mean of carbon dioxide uptake in grass plants") %>%  
    set_header_labels(conc = "Concentration") %>%  
    autofit() %>%  
    width(width = c(1, 1, 1))  
  zz  
}
```

---

as_grouped_data	<i>grouped data transformation</i>
-----------------	------------------------------------

---

### 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](#)

### Examples

```
# as_grouped_data -----
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
data_co2 <- as_grouped_data(x = data_co2, groups = c("Treatment"))
data_co2
```

---

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

**Note**

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\\_chunk](#), [hyperlink\\_text](#), [minibar](#)

**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")

autofit(myft)
```

---

as_paragraph	<i>concatenate chunks in a flextable</i>
--------------	------------------------------------------

---

**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.

**See Also**

[as\\_chunk](#), [minibar](#), [as\\_image](#), [hyperlink\\_text](#)

**Examples**

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

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

autofit(myft)
```

---

autofit

*Adjusts cell widths and heights*

---

**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	<i>add flextable into a Word document</i>
--------------------	-------------------------------------------

---

### 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")
```

---

<b>bold</b>	<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 requires careful settings to avoid overlapping borders.

**See Also**

Other borders management: [border\\_inner\\_h](#), [border\\_inner\\_v](#), [border\\_outer](#), [border\\_remove](#), [hline\\_bottom](#), [hline\\_top](#), [hline](#), [vline\\_left](#), [vline\\_right](#), [vline](#)

**Examples**

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

---

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 defined by a call to <a href="#">fp_border</a>
part	partname of the table (one of 'all', 'body', 'header', 'footer')

### See Also

Other borders management: [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 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 defined by a call to <a href="#">fp_border</a>
part	partname of the table (one of 'all', 'body', 'header', 'footer')

**See Also**

Other borders management: [border\\_inner\\_h](#), [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 defined by a call to <a href="#">fp_border</a>
part	partname of the table (one of 'all', 'body', 'header', 'footer')

**See Also**

Other borders management: [border\\_inner\\_h](#), [border\\_inner\\_v](#), [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

### See Also

Other borders management: [border\\_inner\\_h](#), [border\\_inner\\_v](#), [border\\_outer](#), [border](#), [hline\\_bottom](#), [hline\\_top](#), [hline](#), [vline\\_left](#), [vline\\_right](#), [vline](#)

### Examples

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

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



---

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

---

**Description**

Format character columns in a flextable.

**Usage**

```
colformat_char(x, ...)  
  
## S3 method for class 'flextable'  
colformat_char(x, col_keys, na_str = "",  
  prefix = "", suffix = "", ...)
```

**Arguments**

x	a flextable object
...	additional arguments, unused
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\\_lgl](#), [colformat\\_num](#)

**Examples**

```
dat <- iris  
ft <- flextable(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.

**Usage**

```
colformat_int(x, ...)

## S3 method for class 'flextable'
colformat_int(x, col_keys, big.mark = ",",
  na_str = "", prefix = "", suffix = "", ...)
```

**Arguments**

x	a flextable object
...	additional arguments, unused
col_keys	names of the colkeys
big.mark	see <a href="#">formatC</a>
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\\_lgl](#), [colformat\\_num](#)

**Examples**

```
dat <- mtcars

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

---

colformat_lgl	<i>format logical columns</i>
---------------	-------------------------------

---

**Description**

Format logical columns in a flextable.

**Usage**

```
colformat_lgl(x, ...)

## S3 method for class 'flextable'
colformat_lgl(x, col_keys, true = "true",
  false = "false", na_str = "", prefix = "", suffix = "", ...)
```

**Arguments**

x	a flextable object
...	additional arguments, unused
col_keys	names of the colkeys
false, true	string to be used for logical columns
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](#), [colformat\\_num](#)

**Examples**

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

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

---

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

---

**Description**

Format numeric columns in a flextable.

**Usage**

```
colformat_num(x, ...)

## S3 method for class 'flextable'
colformat_num(x, col_keys, big.mark = ",",
  digits = 2, na_str = "", prefix = "", suffix = "", ...)
```

**Arguments**

x	a flextable object
...	additional arguments, unused
col_keys	names of the colkeys
big.mark, digits	see <a href="#">formatC</a>
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](#), [colformat\\_lgl](#)

**Examples**

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

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")
```

---

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

---

**Description**

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

**Usage**

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

**Arguments**

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

**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)
```

---

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	<i>Get flextable dimensions</i>
---------------	---------------------------------

---

**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	<i>Calculate pretty dimensions</i>
------------	------------------------------------

---

**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 <a href="#">fp_text</a>
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.

**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
```



---

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

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

---

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.

**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**

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.

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

---

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')

**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

*Set flextable rows height*


---

**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)
```

---

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 defined by a call to <a href="#">fp_border</a>
part	partname of the table (one of 'all', 'body', 'header', 'footer')

### See Also

Other borders management: [border\\_inner\\_h](#), [border\\_inner\\_v](#), [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 defined by a call to <a href="#">fp_border</a>
part	partname of the table (one of 'all', 'body', 'header', 'footer')

### See Also

Other borders management: [border\\_inner\\_h](#), [border\\_inner\\_v](#), [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 defined by a call to <a href="#">fp_border</a>
part	partname of the table (one of 'all', 'body', 'header', 'footer')

**See Also**

Other borders management: [border\\_inner\\_h](#), [border\\_inner\\_v](#), [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
```

---

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

---

**Description**

Function used to render flextable in knitr/rmarkdown documents. HTML, Word and PowerPoint 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.

**Word chunk options**

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.

**PowerPoint chunk options**

Position should be defined with options `ft.left` and `ft.top`. These are the top left coordinates of the placeholder that will contain the table. They default to `{r ft.left=1, ft.left=2}`.

**Note**

For Word (docx) output, if pandoc version  $\geq 2.0$  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. Insertion of images is not supported with rmarkdown for Word documents. For PowerPoint (pptx) output, if pandoc version  $< 2.4$  is used, an error will be raised.

**Author(s)**

Maxim Nazarov

---

hyperlink\_text      *chunk of text with hyperlink wrapper*

---

**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, formater = format_fun, url, ...)
```

**Arguments**

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

**See Also**[display](#)Other chunk elements for paragraph: [as\\_chunk](#), [as\\_image](#), [minibar](#)**Examples**

```

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

ft <- flextable(dat)
ft <- compose( x = ft, j = "col",
  value = as_paragraph(
    "This is a link: ",
    hyperlink_text(x = col, url = href ) ) )
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 into a single one</i>
----------	------------------------------------------------

---

**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 = 1:3 )
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.

**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.

**See Also**

Other flextable merging function: [merge\\_at](#), [merge\\_h](#), [merge\\_none](#), [merge\\_v](#)

**Examples**

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

---

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.

### 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	<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.

**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
```

---

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

**Note**

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\\_chunk](#), [as\\_image](#), [hyperlink\\_text](#)

**Examples**

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

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

autofit(myft)
```

---

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 "lrtb", "tbrl", "btlr"
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 = "tbrl", 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\_caption

*set caption*

---

**Description**

set caption value in flextable

**Usage**

```
set_caption(x, caption)
```

**Arguments**

x	flextable object
caption	caption value

**Note**

this will have an effect only when output is HTML.

**Examples**

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

---

set_formatter	<i>set column formatter functions</i>
---------------	---------------------------------------

---

### 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

### set\_formatter\_type

set\_formatter\_type is an helper function to quickly define formatter functions regarding to column types.

### 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.

### 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	<i>Set flextable's headers labels</i>
-------------------	---------------------------------------

---

**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.

**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_2 <- flextable( head( iris ))
ft_2 <- set_header_labels(ft_2,
  values = list(Sepal.Length = "Sepal length",
    Sepal.Width = "Sepal width",
    Petal.Length = "Petal length",
    Petal.Width = "Petal width" ) )

ft_2
```

---

style	<i>Set flextable style</i>
-------	----------------------------

---

**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\_alafoli

*Apply box theme*


---

**Description**

Apply theme box to a flextable

**Usage**

```
theme_alafoli(x)
```

**Arguments**

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

**See Also**

Other flextable theme: [theme\\_booktabs](#), [theme\\_box](#), [theme\\_tron\\_legacy](#), [theme\\_tron](#), [theme\\_vader](#), [theme\\_vanilla](#), [theme\\_zebra](#)

**Examples**

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

---

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

---

**Description**

Apply theme tron to a flextable

**Usage**

```
theme_booktabs(x, fontsize = 11)
```

**Arguments**

x	a flextable object
fontsize	font size in pixel

**See Also**

Other flextable theme: [theme\\_alafoli](#), [theme\\_box](#), [theme\\_tron\\_legacy](#), [theme\\_tron](#), [theme\\_vader](#), [theme\\_vanilla](#), [theme\\_zebra](#)

**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
---	--------------------

**See Also**

Other flextable theme: [theme\\_alafoli](#), [theme\\_booktabs](#), [theme\\_tron\\_legacy](#), [theme\\_tron](#), [theme\\_vader](#), [theme\\_vanilla](#), [theme\\_zebra](#)

**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

**See Also**

Other flextable theme: [theme\\_alafoli](#), [theme\\_booktabs](#), [theme\\_box](#), [theme\\_tron\\_legacy](#), [theme\\_vader](#), [theme\\_vanilla](#), [theme\\_zebra](#)

**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



**See Also**

Other flextable theme: [theme\\_alafoli](#), [theme\\_booktabs](#), [theme\\_box](#), [theme\\_tron](#), [theme\\_vader](#), [theme\\_vanilla](#), [theme\\_zebra](#)

**Examples**

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

---

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

---

**Description**

Apply Sith Lord Darth Vader theme to a flextable

**Usage**

```
theme_vader(x, fontsize = 11)
```

**Arguments**

x	a flextable object
fontsize	font size in pixel

**See Also**

Other flextable theme: [theme\\_alafoli](#), [theme\\_booktabs](#), [theme\\_box](#), [theme\\_tron\\_legacy](#), [theme\\_tron](#), [theme\\_vanilla](#), [theme\\_zebra](#)

**Examples**

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

---

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

---

**Description**

Apply theme vanilla to a flextable

**Usage**

```
theme_vanilla(x)
```

**Arguments**

x                    a flextable object

**See Also**

Other flextable theme: [theme\\_alafoli](#), [theme\\_booktabs](#), [theme\\_box](#), [theme\\_tron\\_legacy](#), [theme\\_tron](#), [theme\\_vader](#), [theme\\_zebra](#)

**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

**See Also**

Other flextable theme: [theme\\_alafoli](#), [theme\\_booktabs](#), [theme\\_box](#), [theme\\_tron\\_legacy](#), [theme\\_tron](#), [theme\\_vader](#), [theme\\_vanilla](#)

**Examples**

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

---

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 defined by a call to <a href="#">fp_border</a>
part	partname of the table (one of 'all', 'body', 'header', 'footer')

**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](#)

**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 defined by a call to <a href="#">fp_border</a>
part	partname of the table (one of 'all', 'body', 'header', 'footer')

### See Also

Other borders management: [border\\_inner\\_h](#), [border\\_inner\\_v](#), [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 defined by a call to <a href="#">fp_border</a>
part	partname of the table (one of 'all', 'body', 'header', 'footer')

**See Also**

Other borders management: [border\\_inner\\_h](#), [border\\_inner\\_v](#), [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

*Delete flextable content*


---

**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 <- flextable(iris)
ft <- width(ft, width = 1)
```

---

xtable_to_flexitable	<i>get a flextable from a xtable object</i>
----------------------	---------------------------------------------

---

**Description**

Get a flextable object from a xtable object.

**Usage**

```
xtable_to_flexitable(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", "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`), 4  
`add_footer_lines` (`add_header_footer`), 4  
`add_footer_row` (`add_header_footer`), 4  
`add_header` (`add_header_footer`), 4  
`add_header_footer`, 4  
`add_header_lines` (`add_header_footer`), 4  
`add_header_row` (`add_header_footer`), 4  
`align`, 5  
`align_nottxt_col` (`align`), 5  
`align_text_col` (`align`), 5  
`as_chunk`, 6, 9, 10, 34, 38  
`as_flextable`, 7, 8  
`as_grouped_data`, 8  
`as_image`, 6, 8, 10, 34, 38  
`as_paragraph`, 6, 8, 9, 9, 21, 33, 38  
`autofit`, 10, 40

`bg`, 11  
`body_add_flextable`, 11  
`body_replace_flextable_at_bkm`  
    (`body_add_flextable`), 11  
`bold`, 12  
`border`, 13, 14–16, 30–32, 51–53  
`border_inner_h`, 13, 14, 15, 16, 30–32, 51–53  
`border_inner_v`, 13, 14, 14, 15, 16, 30–32,  
    51–53  
`border_outer`, 13–15, 15, 16, 30–32, 51–53  
`border_remove`, 13–15, 16, 30–32, 51–53

`colformat_char`, 17, 18–20  
`colformat_int`, 17, 17, 19, 20  
`colformat_lgl`, 17, 18, 18, 20  
`colformat_num`, 17–19, 19  
`color`, 20  
`compose`, 3, 6, 8, 9, 21, 33, 38

`delete_part`, 21  
`dim.flextable`, 22  
`dim_pretty`, 22, 40  
`display`, 23, 34

`empty_blanks`, 24

`fix_border_issues`, 25  
`flextable`, 3, 25, 54  
`flextable-package`, 3  
`font`, 26  
`fontsize`, 27  
`footers` (`set_header_footer_df`), 44  
`footers_flextable_at_bkm`, 28  
`format.flextable`, 28  
`formatC`, 18, 19  
`fp_border`, 14, 15, 30–32, 51–53  
`fp_text`, 6, 23, 33

`headers` (`set_header_footer_df`), 44  
`headers_flextable_at_bkm`, 29  
`height`, 29, 40  
`height_all` (`height`), 29  
`hline`, 13–16, 30, 31, 32, 51–53  
`hline_bottom`, 13–16, 30, 31, 32, 51–53  
`hline_top`, 13–16, 30, 31, 31, 51–53  
`htmltools_value`, 32  
`hyperlink_text`, 6, 9, 10, 33, 38

`italic`, 34

`knit_print.flextable`, 41  
`knit_print.flextable` (`htmltools_value`),  
    32

`merge_at`, 35, 36–38  
`merge_h`, 35, 35, 36–38  
`merge_h_range`, 35, 36, 36, 37, 38  
`merge_none`, 35, 36, 37, 38  
`merge_v`, 35–37, 37  
`minibar`, 6, 9, 10, 34, 38

`padding`, 39  
`ph_with_flextable`, 40  
`ph_with_flextable_at`  
    (`ph_with_flextable`), 40

`print.flextable`, 41

`regulartable` (`flextable`), 25

`rotate`, 41

`set_caption`, 42

`set_footer_df` (`set_header_footer_df`), 44

`set_formatter`, 43

`set_formatter_type` (`set_formatter`), 43

`set_header_df` (`set_header_footer_df`), 44

`set_header_footer_df`, 44

`set_header_labels`, 45

`style`, 45

`theme_alafoli`, 46, 47–50

`theme_booktabs`, 46, 47, 47, 48–50

`theme_box`, 46, 47, 47, 48–50

`theme_tron`, 46, 47, 48, 49, 50

`theme_tron_legacy`, 46–48, 48, 49, 50

`theme_vader`, 46–49, 49, 50

`theme_vanilla`, 46–50, 50

`theme_zebra`, 46–50, 50

`vline`, 13–16, 30–32, 51, 52, 53

`vline_left`, 13–16, 30–32, 51, 52, 53

`vline_right`, 13–16, 30–32, 51, 52, 52

`void`, 53

`width`, 40, 54

`xtable_to_flextable`, 54