

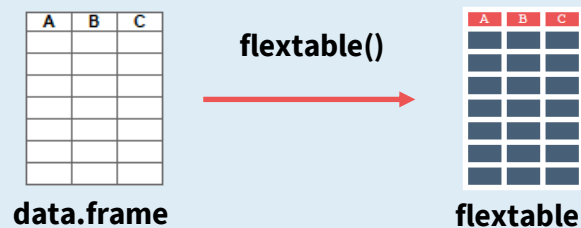
# Tabular reporting with *flextable* : : CHEAT SHEET



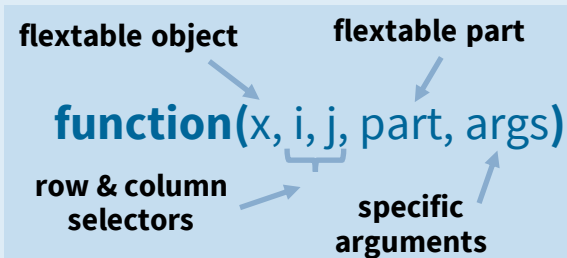
## Basics

The **flextable** package provides a framework for **easily create tables for reporting and publications**.

Functions are provided to let users create tables, modify, format and define their content.



## GENERAL FUNCTION'S STRUCTURE



## TABLE PARTS AND THEIRS DEFAULT VALUES

A	B	C

**header** : colnames

**body** : data

**footer** : empty

all

## Selectors

**i**: row selector

**j**: column selector

### FORMULA

**i** = ~ col %in% "xxx"

col : column name

xxx : value

**j** = ~ col1 + col2

col\* : column name

### CHARACTER VECTOR

**j** = c("col1", "col2")

col\* : column name

### INTEGER VECTOR

**i** = 1:3, **j** = 1:3

### LOGICAL VECTOR

**i** = c(TRUE, FALSE), **j** = c(TRUE, FALSE)

## Format

**GENERAL** `ft <- flextable(data)`

**get\_flextable\_defaults()** : get flextable defaults formatting properties

**set\_flextable\_defaults()** : modify flextable defaults formatting properties

**init\_flextable\_defaults()** : re-init all values with the package defaults

**style(pr\_t, pr\_p, pr\_c)** : modify flextable text, paragraphs and cells formatting properties (needs officer package)

**pr\_t**: object of class fp\_text

**pr\_p**: object of class fp\_par

**pr\_c**: object of class fp\_cell

### TEXT

`font(ft, fontname = "Brush Script MT")`

`fontsize(ft, size = 7)`

`italic(ft, italic = TRUE)`

`bold(ft, bold = TRUE)`

`color(ft, color = "#eb5555")`

`highlight(ft, color = "yellow")`

`rotate(ft, rotation = "tbrl")`

### CELL

`align(ft, align = "center")`

`valign(ft, valign = "top")`

`padding(ft, padding = 10)`

`bg(ft, bg = "#475f77")`

`line_spacing(ft, space = 1.6)`

**THEME** `ft <- theme_*(ft)`

A	B	C
a	b	c
a	b	c
a	b	c

alafoli(ft)

A	B	C
a	b	c
a	b	c
a	b	c

booktabs(ft)

A	B	C
a	b	c
a	b	c
a	b	c

box(ft)

A	B	C
a	b	c
a	b	c
a	b	c

tron(ft)

A	B	C
a	b	c
a	b	c
a	b	c

tron\_legacy(ft)

A	B	C
a	b	c
a	b	c
a	b	c

vader(ft)

A	B	C
a	b	c
a	b	c
a	b	c

vanilla(ft)

A	B	C
a	b	c
a	b	c
a	b	c

zebra(ft)

### BORDER

`brdr <- fp_border(color = "#eb5555", width = 1.5)`

`border_outer(ft, border = brdr)`

`border_inner(ft, border = brdr)`

`border_inner_v(ft, border = brdr)`

`border_inner_h(ft, border = brdr)`

`border_remove(ft)`

`vline_left(ft, border = brdr)`

`vline_right(ft, border = brdr)`

`hline_top(ft, border = brdr)`

`hline_bottom(ft, border = brdr)`

`vline(ft, j = 1:2, border = brdr)`

`hline(ft, i = 1:2, border = brdr)`

## Layout

### HEADER AND FOOTER

#### COLWIDTHS

`add_header_row(ft, values = c("a", "b", "c"), colwidths = c(1, 1, 1), top = FALSE)`

`add_footer_row(ft, values = c("", "", ""), colwidths = c(1, 1, 1))`

#### IN LINE

`add_header_lines(ft, values = "line", top = FALSE)`

`add_footer_lines(ft, values = "line")`

#### COLNAME

`add_header(ft, A = "a", B = "b", top = FALSE)`

`add_body(ft, A = "a", B = "b", C = "")`

`add_footer(ft, A = "", B = "")`

#### GENERAL

`set_header_labels(ft, A = "Aaa", B = "Bbb", C = "Ccc")`

`delete_part(ft, part = "body")`

## Officer

**fp\_text()** : Text formatting properties

color, font.size, bold, italic, underlined, font.family, vertical.align, shading.color

**fp\_par()** : Paragraph formatting properties

text.align, padding, line\_spacing, border, shading.color, padding.bottom, padding.top, padding.left, padding.right, border.bottom, border.left, border.top, border.right

**fp\_cell()** : Cell formatting properties

border, border.bottom, border.left, border.top, border.right, vertical.align, margin, margin.bottom, margin.top, margin.left, margin.right, background.color, text.direction

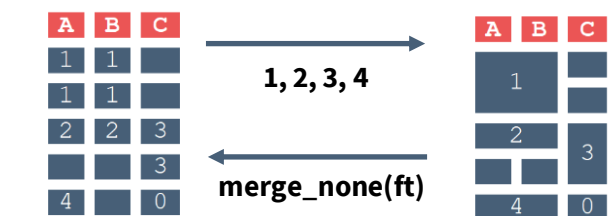
**fp\_border()**: border properties object

color, style, width

**update(x, args)**: update an object of class fp\_\*



### CELL MERGING



1: `merge_at(ft, i = 1:2, j = 1:2)`

2: `merge_h(ft)`

3: `merge_v(ft)`

4: `merge_h_range(ft, i = ~ C %in% "0", j1 = "A", j2 = "B")`

**fix\_border\_issues(ft)**: fix border issues when cell are merged

### CAPTIONS & FOOTNOTES

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

`footnote(ft, j = 1, value = as_paragraph(c("footnote 1")), ref_symbols = c("1"), part = "header")`

# Tabular reporting with *flextable* : : CHEAT SHEET

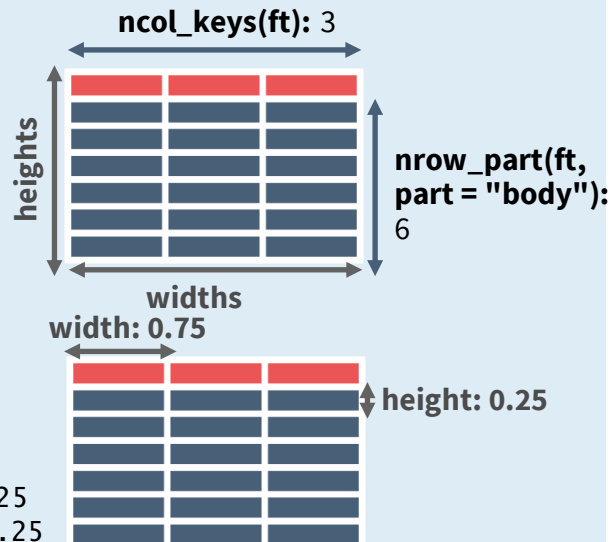


## Table size

```
ft <- flextable(data)
```

```
flextable_dim(ft):
```

```
$widths
[1] 2.25
$heights
[1] 1.75
$aspect_ratio
[1] 0.78
```



```
dim(ft):
```

```
$widths
A B C
0.75 0.75 0.75
$heights
[1] 0.25 0.25 0.25
0.25 0.25 0.25 0.25
```

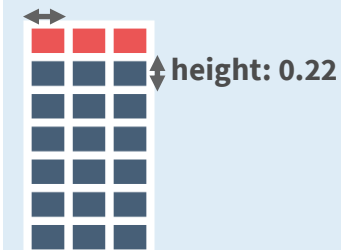
```
dim_pretty(ft):
```

```
$widths
[1] 0.22 0.22 0.22
$heights
[1] 0.22 0.22 0.22 0.22 0.22 0.22 0.22
```

```
autofit(ft, add_w = w, add_h = h)
```

```
w = 0, h = 0
```

```
width: 0.22
```



```
w = 0.2, h = 0
```

```
width: 0.42
```



```
width(ft, i = 1, width = 0.5)
```

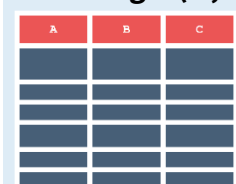
```
width: 0.5 width: 0.75
```



```
ft <- hrule(ft, rule = "exact",
part = "header")
height(ft, height = 0.40, part =
"header")
```



```
ft <- height(ft, i = 1, height = 0.40, part = "body")
ft <- height(ft, i = 4, height = 0.30, part = "body")
```



```
ft <- hrule(ft, rule = "auto", part = "header")
ft <- hrule(ft, i = 1, rule = "exact", part =
"body"): size exactly at 0.4
ft <- hrule(ft, i = 4, rule = "atleast", part =
"body"): size atleast at 0.3
```

## Cell content

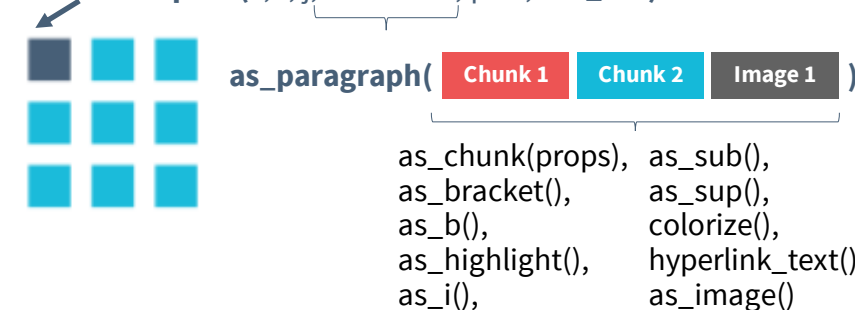
### SIMPLE FORMATTING

args	colformat_char	colformat_date	colformat_datetime	colformat_image	colformat_double	colformat_int	colformat_lgl	colformat_num
x	✓	✓	✓	✓	✓	✓	✓	✓
i	✓	✓	✓	✓	✓	✓	✓	✓
j	✓	✓	✓	✓	✓	✓	✓	✓
na_str	✓	✓	✓	✓	✓	✓	✓	✓
prefix	✓	✓	✓	✓	✓	✓	✓	✓
suffix	✓	✓	✓	✓	✓	✓	✓	✓
big.mark					✓	✓		✓
decimal.mark					✓			✓
fnt_date		✓						
fnt_datetime			✓					
width				✓				
true							✓	
digits					✓			
height				✓				
false							✓	

### MULTI CONTENT

#### FUNCTION COMPOSE

```
compose(x, i, j, value = ..., part, use_dot)
```



**use\_dot():** by default use\_dot=FALSE; if use\_dot=TRUE, value is evaluated within a data.frame augmented of a column named . containing the jth column

```
ft <- flextable(data)
```

```
ft <- compose(ft, value = as_paragraph(
```

```
  as_chunk("chunk"),
  as_bracket("bracket")
  as_b("bold"),
  as_highlight("highlight", color = "yellow")
  as_i("italic"),
  as_sub("sub"),
  as_sup("sup"),
  colorize("colorize", color = "#eb5555"),
  hyperlink_text(hyperlink, url = "http://link"),
  as_image(src, width = 0.2, height = 0.2)))
```

chunk  
(bracket)  
**bold**  
highlight  
italic  
sub  
sup  
colorize  
hyperlink

## Rendering

**flextable** default format is **HTML** output printed in the rstudio viewer pane.

**flextable** objects can be rendered in **HTML** format, **Microsoft Word**, **Microsoft PowerPoint** and **PDF**.



### SIMPLE EXPORT

```
save_as_html(ft, "ft.html")
save_as_docx(ft, "ft.docx")
save_as_pptx(ft, "ft.pptx")
save_as_image(ft, "ft.png")
```

### INTERACTIVE SESSION

```
print(ft, preview = "docx")
print(ft, preview = "pptx")
```

### RMARKDOWN DOCUMENTS

```
```{r}
library(flextable)
ft <- flextable(ft)
ft
```
```

```
LOOPING IN RMARKDOWN WITH FOR
flextable_to_rmd(ft)
```

### WITH OFFICER

```
ph_with(ppt, value = ft) (PowerPoint)
ppt : an rpptx object
body_add_flextable(value = ft) (Word)
```

### IN SHINY

```
library(shiny)
library(flextable)
ft <- flextable(data)
# In UI
uiOutput("ft")
```

```
# In server
output$ft <- renderUI({
  htmltools_value(ft)
})
```