

Package ‘huxtable’

July 3, 2018

Type Package

Title Easily Create and Style Tables for LaTeX, HTML and Other Formats

Version 4.0.1

Author David Hugh-Jones [aut, cre]

Maintainer David Hugh-Jones <davidhughjones@gmail.com>

Description Like 'xtable', creates styled tables. Export to HTML, LaTeX, 'Word', 'Excel', 'PowerPoint' and other formats. Simple, modern interface to manipulate borders, size, position, captions, colours, text styles and number formatting. Table cells can span multiple rows and/or columns. Includes a 'huxreg' function for creation of regression tables, and 'quick_*' one-liners to print data to a new document.

License MIT + file LICENSE

URL <https://hughjonesd.github.io/huxtable>

BugReports <https://github.com/hughjonesd/huxtable/issues>

Imports assertthat, dplyr, glue, htmltools, knitr, rmarkdown, stringr, tibble, tidyselect, utils

Suggests broom, covr, crayon, devtools, flextable, ggplot2, lazyeval, lme4, lmtest, nlme, nnet, officer, openxlsx, psych, testthat

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-03 13:50:03 UTC

R topics documented:

huxtable-package	3
add_colnames	4
add_footnote	5
align	6
as_FlexTable	7
as_Workbook	8
background_color	9
bold	10
caption	11
caption_pos	12
cbind.huxtable	13
colspan	14
col_width	15
escape_contents	16
every	17
filter	18
final	18
font	19
font_size	20
get_default_properties	21
guess_knitr_output_format	21
height	22
huxreg	23
huxtable	24
hux_logo	26
insert_column	27
is_a_number	28
knit_print.data.frame	28
knit_print.huxtable	29
label	30
latex_float	31
left_border	31
left_border_color	33
left_padding	34
mutate.huxtable	35
na_string	36
number_format	37
position	39
print.huxtable	40
print_html	41
print_latex	42
print_md	42
print_screen	43
quick-output	45
report_latex_dependencies	46
rotation	46

rowspan	47
rowspecs	48
row_height	49
sanitize	50
set_all_borders	51
set_all_border_colors	51
set_cell_properties	52
set_default_properties	53
set_outer_borders	54
t.huxtable	54
tabular_environment	55
text_color	56
themes	57
valign	58
where	59
width	59
wrap	60
[.huxtable	61

Index **63**

huxtable-package *Huxtable: simply create LaTeX and HTML tables*

Description

Huxtable is a package for creating HTML and LaTeX tables. It provides similar functionality to xtable, with a simpler interface.

Details

To create a huxtable object, use `huxtable()` or `as_huxtable()`.

For more information, see [the website](#) or read the vignette with `vignette('huxtable')`.

Package options

- `options('huxtable.add_colnames')` sets the default value for `add_colnames` in `huxtable()` and `as_huxtable()`. If it is unset, `add_colnames` defaults to `FALSE`; in a future release, the default will become `TRUE`.
- `options('huxtable.print')` sets the print method for huxtable objects. See `print.huxtable()`.
- `options('huxtable.color_screen')`. If `TRUE` and package `crayon` is available, huxtables will be printed in color on screen.
- `options('huxtable.knit_print_df')`. If `TRUE` (the default), data frames in knitr will be pretty-printed using huxtable.
- `options('huxtable.knit_print_df_theme')`. A one-argument function applied to theme the huxtableized data frame before printing in knitr. Defaults to `theme_plain()`.

- `options('huxtable.autoformat')` sets the default value for `autoformat` in `huxtable()` and `as_huxtable()`. It defaults to `TRUE`.
- `options('huxtable.autoformat_number_format')` and `options('huxtable.autoformat_align')` are lists. The list names are base R classes. `huxtable()` with `autoformat = TRUE` will set `number_format()` and `align()` for data columns according to the corresponding list values.

For example, to center-align Date objects you could set `"huxtable.autoformat_align"` to something like `list(..., Date = "center", ...)`.

add_colnames	<i>Add column or row names</i>
--------------	--------------------------------

Description

Add a first row of column names, or a first column of row names, to the huxtable.

Usage

```
add_colnames(ht, ...)

## S3 method for class 'huxtable'
add_colnames(ht, rowname = NULL, ...)

add_rownames(ht, ...)

## S3 method for class 'huxtable'
add_rownames(ht, colname = "rownames",
  preserve_rownames = TRUE, ...)
```

Arguments

<code>ht</code>	A huxtable.
<code>...</code>	Arguments passed to methods.
<code>rowname</code>	Optional row name for the new row of column names.
<code>colname</code>	Column name for the new column of row names.
<code>preserve_rownames</code>	Preserve existing row names.

Details

Note that `add_colnames` will change the mode of all columns to character. Also note that it will move your rows down by one: what was row 1 will now be row 2, and the column names will now be row 1.

`add_colnames` preserves column names. `add_rownames` only preserves them if asked to.

Value

The modified object.

Examples

```
ht <- huxtable(a = 1:5, b = 1:5)
add_rownames(ht)
add_colnames(ht)
add_rownames(add_colnames(ht)) # Out by 1
add_colnames(add_rownames(ht)) # Better
add_colnames(add_rownames(ht, '')) # Alternatively
```

add_footnote	<i>Add a row with a footnote</i>
--------------	----------------------------------

Description

This adds a single row at the bottom. The first cell contains the footnote; it spans all table columns and has an optional border above.

Usage

```
add_footnote(ht, text, border = 0.8, ...)
```

Arguments

ht	A huxtable.
text	Text for the footnote.
border	Width of the footnote's top border. Set to 0 for no border.
...	Other properties, passed to set_cell_properties() for the footnote cell.

Value

The modified huxtable

Examples

```
ht <- hux(a = 1:5, b = 1:5, d = 1:5)
ht <- add_footnote(ht, '* this is a footnote')
ht
```

align	<i>Alignment</i>
-------	------------------

Description

Functions to get or set the alignment property of huxtable table cells.

Usage

```
align(ht)
align(ht) <- value
set_align(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A character vector or matrix which may be 'left', 'center', 'right', NA or a single character. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Details

This sets the horizontal alignment of the cell. If value is a single character (e.g. a decimal point), then the cell is aligned on this character.

Value

For align, the align attribute. For set_align, the ht object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
align(ht) <- 'right'
align(ht)

ht <- huxtable(a = 1:3, b = 3:1)
set_align(ht, 'right')
set_align(ht, 1:2, 1, 'right')
set_align(ht, 1:2, 1:2, c('right', 'left'), byrow = TRUE)
set_align(ht, where(ht == 1), 'right')
```

`as_FlexTable`*Convert a huxtable for Word/Powerpoint*

Description

Huxtables can be converted to `flextable::flextable()` objects, for use in Word and Powerpoint documents.

Usage

```
as_FlexTable(x, ...)
```

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

```
## S3 method for class 'huxtable'
```

```
as_flextable(x, colnames_to_header = FALSE, ...)
```

Arguments

`x` A huxtable.

`...` Not used.

`colnames_to_header`

Use huxtable column names as the header. If FALSE, the flextable will contain only a body and no header.

Details

Note: you can't use flextable Word output within rmarkdown. Instead you have to write the Word file yourself. See `officer::read_docx()`.

`as_FlexTable` is deprecated and calls `as_flextable` with a warning.

Properties are supported, with the following exceptions:

- Rotation of 0, 90 or 270 is supported.
- Non-numeric column widths and row heights are not supported.
- Table height, wrap, captions and table position are not supported.

Value

an object of class `flextable`.

Challenge

Try to say `as_flextable.huxtable` ten times without pausing.

Examples

```
ht <- hux(a = 1:3, b = 1:3)
ft <- as_flextable(ht)
## Not run:
my_doc <- officer::read_docx()
my_doc <- flextable::body_add_flextable(my_doc, ft)
print(my_doc, target = "path/to/my_doc.docx")

## End(Not run)
```

as_Workbook

Convert a huxtable for Excel

Description

If the `openxlsx` package is installed, Huxtables can be converted to `openxlsx::openxlsx()` Workbook objects, for use in Excel documents.

Usage

```
as_Workbook(ht, ...)

## S3 method for class 'huxtable'
as_Workbook(ht, Workbook = NULL, sheet = "Sheet 1",
  write_caption = TRUE, ...)
```

Arguments

ht	A huxtable.
...	Not used.
Workbook	An existing Workbook object. By default, a new workbook will be created.
sheet	Name for the worksheet where the huxtable will be created.
write_caption	If TRUE, print any caption in the row above or below the table.

Details

Use `openxlsx::saveWorkbook()` to save the resulting object to an Excel file.

Properties are supported with the following exceptions:

- Non-numeric column widths and row heights, table width and height.
- Decimal padding.
- Cell padding.
- Table position.

Huxtable tries to guess appropriate widths and height for rows and columns; numeric `width()` and `height()` are treated as scaling factors.

Contents are only stored as numbers if a whole column is numeric as defined by `is_a_number()`; otherwise they are stored as text.

Value

An object of class Workbook.

Examples

```
ht <- hux(a = 1:3, b = 1:3)
wb <- as_Workbook(ht)
## Not run:
openxlsx::saveWorkbook(wb, "my-excel-file.xlsx")

## End(Not run)
```

background_color	<i>Background color</i>
------------------	-------------------------

Description

Functions to get or set the background color property of huxtable table cells.

Usage

```
background_color(ht)
background_color(ht) <- value
set_background_color(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A character vector or matrix of valid R color names. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Value

For background_color, the background_color attribute. For set_background_color, the ht object.

See Also

Other formatting functions: [bold](#), [font_size](#), [font](#), [na_string](#), [number_format](#), [text_color](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
background_color(ht) <- grey(.95)
background_color(ht)
```

```
ht <- huxtable(a = 1:3, b = 3:1)
set_background_color(ht, grey(.95))
set_background_color(ht, 1:2, 1, grey(.95))
set_background_color(ht, 1:2, 1:2, c(grey(.95), 'yellow'), byrow = TRUE)
set_background_color(ht, where(ht == 1), grey(.95))
```

bold*Cell text style*

Description

Functions to get or set the cell text style property of huxtable table cells.

Usage

```
bold(ht)
bold(ht) <- value
set_bold(ht, row, col, value, byrow = FALSE)

italic(ht)
italic(ht) <- value
set_italic(ht, row, col, value, byrow = FALSE)
```

Arguments

<code>ht</code>	A huxtable.
<code>value</code>	A logical vector or matrix Set to NA to reset to the default.
<code>row</code>	A row specifier. See rowspecs for details.
<code>col</code>	An optional column specifier.
<code>byrow</code>	If TRUE, fill in values by row rather than by column.

Value

For `bold`, the bold attribute. For `set_bold`, the ht object.

See Also

Other formatting functions: [background_color](#), [font_size](#), [font](#), [na_string](#), [number_format](#), [text_color](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
bold(ht) <- TRUE
bold(ht)
print_screen(ht)

ht <- huxtable(a = 1:3, b = 3:1)
set_bold(ht, TRUE)
set_bold(ht, 1:2, 1, TRUE)
set_bold(ht, 1:2, 1:2, c(TRUE, FALSE), byrow = TRUE)
set_bold(ht, where(ht == 1), TRUE)
```

caption

Caption

Description

Functions to get or set the table-level caption property of a huxtable.

Usage

```
caption(ht)
caption(ht) <- value
set_caption(ht, value)
```

Arguments

ht A huxtable.
value A length-one character vector. Set to NA for no caption.

Details

Captions are not escaped. See the example for a workaround.

Value

For `caption`, the caption attribute. For `set_caption`, the ht object.

See Also

[caption_pos\(\)](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
caption(ht) <- 'An example table'
caption(ht)
print_screen(ht)
ht <- hux(a = 1:2, b = 1:2)
caption(ht) <- sanitize('Make $$$ with us', type = 'latex') # escape caption characters
```

caption_pos

Caption position

Description

Functions to get or set the table-level caption position property of a huxtable.

Usage

```
caption_pos(ht)
caption_pos(ht) <- value
set_caption_pos(ht, value)
```

Arguments

ht	A huxtable.
value	A length-one character vector, one of 'top', 'bottom', 'topleft', 'topcenter', 'topright', 'bottomleft', 'bottomcenter', 'bottomright', or NA for the default.

Details

If caption_pos is 'top' or 'bottom', then the horizontal position ('left', 'center' or 'right') will be determined by the huxtable's [position\(\)](#).

Value

For caption_pos, the caption_pos attribute. For set_caption_pos, the ht object.

See Also

[caption\(\)](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
caption_pos(ht) <- 'bottom'
caption_pos(ht)
```

cbind.huxtable	<i>Combine rows or columns</i>
----------------	--------------------------------

Description

Combine rows or columns

Usage

```
## S3 method for class 'huxtable'
cbind(..., deparse.level = 1, copy_cell_props = TRUE)

## S3 method for class 'huxtable'
rbind(..., deparse.level = 1, copy_cell_props = TRUE)
```

Arguments

... Vectors, matrices, data frames or huxtables.
 deparse.level Passed to `cbind.data.frame()`.
 copy_cell_props Cell properties to copy from neighbours (see below).

Details

Table properties will be taken from the first argument which is a huxtable. So will row properties (for `cbind`) and column properties (for `rbind`).

If some of the inputs are not huxtables, and `copy_cell_props` is a character vector of cell properties, then for `rbind`, the named cell properties and row heights will be copied to non-huxtables. For `cbind`, the named cell properties and column widths will be copied. Objects on the left or above get priority over those on the right or below.

If `copy_cell_props` is `TRUE`, the default set of cell properties (everything but `colspan` and `rowspan`) will be copied.

If `copy_cell_props` is `FALSE`, cells from non-huxtable objects will get the default properties.

Value

A huxtable.

Examples

```
ht1 <- hux(a = 1:3, b = 4:6)
ht2 <- hux(d = letters[1:3], e = letters[4:6])
bold(ht1)[1,] <- TRUE
bold(ht2) <- TRUE
vec <- LETTERS[1:3]

ht_out <- cbind(ht1, vec, ht2)
```

```
ht_out
bold(ht_out)
bold(cbind(ht1, vec, ht2, copy_cell_props = FALSE))
```

colspan	<i>Column span</i>
---------	--------------------

Description

Functions to get or set the column span property of huxtable table cells.

Usage

```
colspan(ht)
colspan(ht) <- value
set_colspan(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	An integer vector or matrix of integers. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Value

For colspan, the colspan attribute. For set_colspan, the ht object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
colspan(ht)[1, 1] <- 2
colspan(ht)
print_screen(ht)
```

col_width	<i>Column widths</i>
-----------	----------------------

Description

Functions to get or set the column widths property of huxtable cols.

Usage

```
col_width(ht)
col_width(ht) <- value
set_col_width(ht, col, value)
```

Arguments

ht	A huxtable.
value	A vector. If numeric, they are treated as proportions of the table width. If character, they must be valid CSS or LaTeX lengths.
col	A col specifier. See rowspecs for details.

Details

In LaTeX, if you specify a column width, but set wrap to FALSE and have cells which overrun, then you may have problems with table position and with background colours in other cells. The workaround is to adjust the width, so that your cells no longer overrun.

Value

For col_width, the col_width attribute. For set_col_width, the ht object.

See Also

Other row/column heights: [row_height](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
col_width(ht) <- c(.2, .8)
col_width(ht)
print_screen(ht)
```

escape_contents	<i>Escape cell contents</i>
-----------------	-----------------------------

Description

Functions to get or set the escape cell contents property of huxtable table cells.

Usage

```
escape_contents(ht)
escape_contents(ht) <- value
set_escape_contents(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A logical vector or matrix. If TRUE, cell contents will be HTML or LaTeX escaped. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Value

For `escape_contents`, the `escape_contents` attribute. For `set_escape_contents`, the `ht` object.

Examples

```
ht <- huxtable(Exponent = 2:4, Example = paste0('$x^', 2:4, '$'))
escape_contents(ht)[,2] <- FALSE

ht <- huxtable(a = 1:3, b = 3:1)
ht2 <- set_escape_contents(ht, TRUE)
escape_contents(ht2)
ht3 <- set_escape_contents(ht, 1:2, 1, TRUE)
escape_contents(ht3)
ht4 <- set_escape_contents(ht, 1:2, 1:2, c(TRUE, FALSE), byrow = TRUE)
escape_contents(ht4)
ht5 <- set_escape_contents(ht, where(ht == 1), TRUE)
escape_contents(ht5)
```

every *Return every n row or column numbers*

Description

This is a convenience function to use in row or column specifications. In this context, `every(n, from)` will return `from, from + n, ...`, up to the number of rows or columns of the huxtable. `evens` and `odds` return even and odd numbers, i.e. they are equivalent to `every(2, 2)` and `every(2, 1)` respectively. `everywhere` returns all rows or columns, equivalently to `every(1)`.

Usage

```
every(n = 1, from = n)

everywhere(ht, dimension)

evens(ht, dimension)

odds(ht, dimension)
```

Arguments

<code>n</code>	A number (at least 1)
<code>from</code>	A number (at least 1)
<code>ht</code>	An object with a <code>dim</code> attribute like a matrix or data frame.
<code>dimension</code>	Number of the dimension to use.

Details

Technically, `every` returns a 2-argument function which can be called like `f(ht, dimension)`. See [rowspecs](#) for details.

Examples

```
ht <- huxtable(a = 1:10, b = 1:10)
ht <- set_background_color(ht, every(3), everywhere, 'wheat')
background_color(ht)
ht <- set_align(ht, evens, 1:2, 'right')
ht <- set_align(ht, odds, 1:2, 'center')
align(ht)
```

filter	<i>Pointless documentation</i>
--------	--------------------------------

Description

This pointless piece of documentation exists to satisfy R CMD check, due to a complicated and boring issue involving `stats::filter`, `dplyr::filter` and the mysterious workings of the NAMES-SPACE file. See [mutate.huxtable\(\)](#) for details about using `dplyr` with `huxtable`.

Arguments

.data	Data
...	Other args

final	<i>Return the last n rows or columns</i>
-------	--

Description

This is a convenience function to use in row and column specifications. In that context, it returns the last n row or column numbers of the `huxtable`.

Usage

```
final(n = 1)
```

Arguments

n	Number of rows to return.
---	---------------------------

Details

Technically, `final` returns a two-argument function - see [rowspecs](#) for more details.

Examples

```
ht <- hux(a = 1:5, b = 1:5, d = 1:5, e = 1:5)
ht <- set_align(ht, final(2), final(1), 'left')
align(ht)
```

```
final(3)(ht, 1) # last 3 rows
final(3)(ht, 2) # last 3 columns
```

font	<i>Font</i>
------	-------------

Description

Functions to get or set the font property of huxtable table cells.

Usage

```
font(ht)
font(ht) <- value
set_font(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A character vector of font names. NB that LaTeX and HTML use different font names. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Value

For font, the font attribute. For set_font, the ht object.

See Also

Other formatting functions: [background_color](#), [bold](#), [font_size](#), [na_string](#), [number_format](#), [text_color](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
font(ht) <- 'times'
font(ht)

ht <- huxtable(a = 1:3, b = 3:1)
ht2 <- set_font(ht, 'times')
font(ht2)
ht3 <- set_font(ht, 1:2, 1, 'times')
font(ht3)
ht4 <- set_font(ht, 1:2, 1:2, c('times', 'arial'), byrow = TRUE)
font(ht4)
ht5 <- set_font(ht, where(ht == 1), 'times')
font(ht5)
```

font_size	<i>Font size</i>
-----------	------------------

Description

Functions to get or set the font size property of huxtable table cells.

Usage

```
font_size(ht)
font_size(ht) <- value
set_font_size(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A numeric vector. This sets the font size in points. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Value

For font_size, the font_size attribute. For set_font_size, the ht object.

See Also

Other formatting functions: [background_color](#), [bold](#), [font](#), [na_string](#), [number_format](#), [text_color](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
font_size(ht) <- 14
font_size(ht)

ht <- huxtable(a = 1:3, b = 3:1)
ht2 <- set_font_size(ht, 14)
font_size(ht2)
ht3 <- set_font_size(ht, 1:2, 1, 14)
font_size(ht3)
ht4 <- set_font_size(ht, 1:2, 1:2, c(14, 12), byrow = TRUE)
font_size(ht4)
ht5 <- set_font_size(ht, where(ht == 1), 14)
font_size(ht5)
```

get_default_properties
Get default hutable properties

Description

Get default hutable properties

Usage

```
get_default_properties(names = NULL)
```

Arguments

names Vector of property names. If NULL, all properties are returned.

Value

List of default properties.

See Also

[set_default_properties\(\)](#)

Examples

```
get_default_properties('bold')  
get_default_properties()
```

guess_knitr_output_format
Guess knitr output format

Description

Convenience function which tries to guess the ultimate output from knitr and rmarkdown.

Usage

```
guess_knitr_output_format()
```

Value

'html', 'latex', or something else. If we are not in a knitr document, returns an empty string.

Examples

```
## Not run:
# in a knitr document
guess_knitr_output_format()

## End(Not run)
```

height	<i>Table height</i>
--------	---------------------

Description

Functions to get or set the table-level table height property of a huxtable.

Usage

```
height(ht)
height(ht) <- value
set_height(ht, value)
```

Arguments

ht	A huxtable.
value	A length-one vector. If numeric, it is treated as a proportion of the containing block height for HTML, or of text height (<code>\textheight</code>) for LaTeX. If character, it must be a valid CSS or LaTeX width. Set to NA for the default, which is to leave height unset.

Value

For `height`, the height attribute. For `set_height`, the ht object.

See Also

Other table measurements: [width](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
height(ht) <- 0.4
height(ht)
```

huxreg *Create a huxtable to display model output*

Description

Create a huxtable to display model output

Usage

```
huxreg(..., error_format = "{std.error}", error_style = c("stderr", "ci",
  "statistic", "pvalue"), error_pos = c("below", "same", "right"),
  number_format = "%.3f", align = ".", pad_decimal = ".",
  ci_level = NULL, tidy_args = NULL, stars = c(`***` = 0.001, `**` = 0.01,
  `*` = 0.05), bold_signif = NULL, borders = 0.4, outer_borders = 0.8,
  note = "{stars}.", statistics = c(N = "nobs", R2 = "r.squared", "logLik",
  "AIC"), coefs = NULL, omit_coefs = NULL)
```

Arguments

...	Models, or a single list of models. Names will be used as column headings.
error_format	How to display uncertainty in estimates. See below.
error_style	Deprecated. One or more of 'stderr', 'ci' (confidence interval), 'statistic' or 'pvalue'.
error_pos	Display uncertainty 'below', to the 'right' of, or in the 'same' cell as estimates.
number_format	Format for numbering. See number_format() for details.
align	Alignment for table cells. Set to a single character to align on this character.
pad_decimal	Deprecated in favour of align.
ci_level	Confidence level for intervals. Set to NULL to not calculate confidence intervals.
tidy_args	List of arguments to pass to broom::tidy() . You can also pass a list of lists; if so, the nth element will be used for the nth column.
stars	Levels for p value stars. Names of stars are symbols to use. Set to NULL to not show stars.
bold_signif	Where p values are below this number, cells will be displayed in bold. Use NULL to turn off this behaviour.
borders	Thickness of inner horizontal borders. Set to 0 for no borders.
outer_borders	Thickness of outer (top and bottom) horizontal borders. Set to 0 for no borders.
note	Footnote for bottom cell, which spans all columns. {stars} will be replaced by a note about significance stars. Set to NULL for no footnote.
statistics	Summary statistics to display. Set to NULL to show all available statistics.
coefs	Display only these coefficients. Overrides <code>omit_coefs</code> .
omit_coefs	Omit these coefficients.

Details

Models must have a `broom::tidy()` method defined, which should return 'term', 'estimate', 'std.error', 'statistic' and 'p.value'. If the tidy method does not have a `conf.int` option, `huxreg` will calculate confidence intervals itself, using a normal approximation.

If `...` has names or contains a single named list, the names will be used for column headings. Otherwise column headings will be automatically created. If the `coef` and/or `statistics` vectors have names, these will be used for row headings. If different values of `coef` have the same name, the corresponding rows will be merged in the output.

Each element of `statistics` should be a column name from `broom::glance()`. You can also use 'nobs' for the number of observations. If `statistics` is `NULL` then all columns from `glance` will be used. To use no columns, set `statistics = character(0)`.

`error_format` is a string to be interpreted by `glue::glue()`. Terms in parentheses will be replaced by computed values. You can use any columns returned by `tidy`: typical columns include `statistic`, `p.value`, `std.error`, as well as `conf.low` and `conf.high` if you have set `ci_level`. For example, to show confidence intervals, you could do `error_format = "{conf.low} to {conf.high}"`

Value

A `huxtable` object.

Examples

```
set.seed(27101975)
dfr <- data.frame(a = rnorm(100), b = rnorm(100))
dfr$y <- dfr$a + rnorm(100)
lm1 <- lm(y ~ a, dfr)
lm2 <- lm(y ~ a + b, dfr)
glm1 <- glm(I(y > 0) ~ a, dfr, family = binomial)
huxreg(lm1, lm2, glm1)
```

huxtable

Create a huxtable

Description

`huxtable`, or `hux`, creates a `huxtable` object.

Usage

```
huxtable(..., add_colnames = getOption("huxtable.add_colnames", FALSE),
  add_rownames = FALSE, autoformat = getOption("huxtable.autoformat", TRUE))
```

```
hux(..., add_colnames = getOption("huxtable.add_colnames", FALSE),
  add_rownames = FALSE, autoformat = getOption("huxtable.autoformat", TRUE))
```

```
as_huxtable(x, ...)
```



```

as_hux(x, ...)

## Default S3 method:
as_huxtable(x,
  add_colnames = getOption("huxtable.add_colnames", FALSE),
  add_rownames = FALSE, autoformat = getOption("huxtable.autoformat", TRUE),
  ...)

is_huxtable(x)

is_hux(x)

```

Arguments

...	For huxtable, named list of values as in <code>data.frame()</code> . For as_huxtable, extra arguments.
add_colnames	If TRUE, add a first row of column names to the huxtable.
add_rownames	If TRUE, add a first column of row names, named 'rownames', to the huxtable.
autoformat	If TRUE, automatically format columns by type. See below.
x	An object to convert to a huxtable.

Details

If you use `add_colnames` or `add_rownames`, be aware that these will shift your rows and columns along by one: your old row/column 1 will now be row/column 2, etc.

`add_colnames` currently defaults to FALSE, but this will change in future. You can set the default globally by setting `options("huxtable.add_colnames")` to TRUE or FALSE.

`as_huxtable` and `as_hux` convert an object to a huxtable. Conversion methods exist for data frames, tables, ftables, matrices and (most) vectors.

Value

An object of class `huxtable`.

Automatic formatting

If `autoformat` is TRUE, then columns will have `number_format()` and `align()` properties set automatically, as follows:

- Integer columns will have `number_format` set to 0.
- Other numeric columns will have `number_format` set to `"%.3g"`.
- All other columns will have `number_format` set to NA (no formatting).
- Integer, Date and date-time (i.e. POSIXct and POSIXlt) columns will be right-aligned.
- Other numeric columns will be aligned on `options("OutDec")`, usually `"."`.
- Other columns will be left aligned.

You can change these defaults by editing `options("huxtable.autoformat_number_format")` and `options("huxtable.autoformat_align")`. See [huxtable-package](#) for more details.

Automatic alignment also applies to column headers if `add_colnames` is TRUE; headers of columns aligned on a decimal point will be right-aligned. Automatic number formatting does not apply to column headers.

Examples

```
ht <- huxtable(column1 = 1:5, column2 = letters[1:5])
dfr <- data.frame(a = 1:5, b = letters[1:5], stringsAsFactors = FALSE)
as_huxtable(dfr)
```

`hux_logo`

Huxtable logo

Description

Huxtable logo

Usage

```
hux_logo(latex = FALSE)
```

Arguments

`latex` Use LaTeX names for fonts.

Value

The huxtable logo.

Examples

```
print_screen(hux_logo())
```

insert_column	<i>Insert a row or column</i>
---------------	-------------------------------

Description

These convenience functions wrap `cbind` or `rbind` for huxtables to insert a single row.

Usage

```
insert_column(ht, ..., after = 0, copy_cell_props = TRUE)
```

```
insert_row(ht, ..., after = 0, copy_cell_props = TRUE)
```

Arguments

<code>ht</code>	A huxtable.
<code>...</code>	Cell contents.
<code>after</code>	Insert the row/column after this position. 0 (the default) inserts as the first row/column.
<code>copy_cell_props</code>	Copy cell properties from the previous row or column (if <code>after > 0</code>). See cbind.huxtable() .

Details

In `insert_column` only, you can use a column name for `after`.

Value

The modified huxtable

Examples

```
ht <- hux(a = 1:5, b = 1:5, c = 1:5)
insert_row(ht, 2.5, 2.5, 2.5, after = 2)
insert_column(ht, 5:1)
insert_column(ht, 5:1, after = 3)
insert_column(ht, 5:1, after = "b")
```

 is_a_number

Does an object look like a number?

Description

A convenience function that returns TRUE if an object either is numeric or can be converted to a number. For data frames, it returns a matrix of the same dimensions as the data frame.

Usage

```
is_a_number(x)
```

Arguments

x An object.

Value

A logical object with the same dimensions as x.

Examples

```
is_a_number(1.0)
is_a_number("1.0")
is_a_number("a")
ht <- hux(a = 1:3, b = 1:3, add_colnames = TRUE)
is_a_number(ht)
```

 knit_print.data.frame *Print data frames in knitr using huxtable*

Description

Print data frames in knitr using huxtable

Usage

```
## S3 method for class 'data.frame'
knit_print(x, options, ...)
```

Arguments

x A huxtable.
 options Not used.
 ... Not used.

Details

huxtable defines a `knit_print` method for `data.frames`. This converts the data frame to a huxtable, with `add_colnames = TRUE`, themes it using `theme_plain()` and prints it. To turn this behaviour off, set `options(huxtable.knit_print_df = FALSE)`. To change the theme, set `options("huxtable.knit_print_df_theme")` to a one-argument function which should return the huxtable.

See Also

Other knit_print: [knit_print.huxtable](#)

Examples

```
## Not run:
# in your knitr document
mytheme <- function (ht) {
  ht <- set_all_borders(ht, 0.4)
  ht <- set_all_border_colors(ht, "darkgreen")
  ht <- set_background_color(ht, evens, odds, "salmon")
  ht
}

options(huxtable.knit_print_df_theme = mytheme)
data.frame(a = 1:5, b = 1:5) # groovy!

## End(Not run)
```

`knit_print.huxtable` *Print a huxtable within knitr*

Description

Print a huxtable within knitr

Usage

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

Arguments

<code>x</code>	A huxtable.
<code>options</code>	Not used.
<code>...</code>	Not used.

Details

knitr calls `knitr::knit_print()` on objects when they are printed in a knitr (or RMarkdown) document. The default method for huxtable objects guesses the appropriate output format and prints itself out appropriately.

See Also

Other knit_print: [knit_print.data.frame](#)

label	<i>Table label</i>
-------	--------------------

Description

Functions to get or set the table-level table label property of a huxtable.

Usage

```
label(ht)
label(ht) <- value
set_label(ht, value)
```

Arguments

ht	A huxtable.
value	A length-one character vector to be used as a table label in LaTeX, or as an ID for the table in HTML. Set to NA to remove any label.

Details

LaTeX table labels typically start with "tab:", and they must do so if you want table numbering in [bookdown](#).

Value

For label, the label attribute. For set_label, the ht object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
label(ht) <- 'tab:mytable'
label(ht)
```

latex_float	<i>Float position for LaTeX</i>
-------------	---------------------------------

Description

Functions to get or set the table-level float position for latex property of a huxtable.

Usage

```
latex_float(ht)
latex_float(ht) <- value
set_latex_float(ht, value)
```

Arguments

ht	A huxtable.
value	A length-one character vector, used by LaTeX for positioning the float. Set to NA for the default, 'h'.

Details

Quick reference: 'h' here, 'h!' definitely here, 't' top of page, 'b' bottom of page, 'p' page of floats. See LaTeX documentation for more details. If you use 'H' (definitely here), you must require the float package.

Value

For latex_float, the latex_float attribute. For set_latex_float, the ht object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
latex_float(ht) <- 'tab:mytable'
latex_float(ht)
```

left_border	<i>Borders</i>
-------------	----------------

Description

Functions to get or set the borders property of huxtable table cells.

Usage

```

left_border(ht)
left_border(ht) <- value
set_left_border(ht, row, col, value, byrow = FALSE)

right_border(ht)
right_border(ht) <- value
set_right_border(ht, row, col, value, byrow = FALSE)

top_border(ht)
top_border(ht) <- value
set_top_border(ht, row, col, value, byrow = FALSE)

bottom_border(ht)
bottom_border(ht) <- value
set_bottom_border(ht, row, col, value, byrow = FALSE)

```

Arguments

ht	A huxtable.
value	A numeric vector or matrix giving border widths in points. Set to 0 for no border. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Details

Currently in LaTeX, all non-zero border widths on a given line must be the same, and vertical border widths can only be present (if `value > 0`) or absent.

Value

For `left_border`, the `left_border` attribute. For `set_left_border`, the `ht` object. Similarly for the other functions.

See Also

[set_all_borders\(\)](#)

Examples

```

ht <- huxtable(a = 1:3, b = 1:3)
left_border(ht) <- 1
left_border(ht)
print_screen(ht)

```



```
ht <- huxtable(a = 1:3, b = 3:1)
set_left_border(ht, 1)
set_left_border(ht, 1:2, 1, 1)
set_left_border(ht, 1:2, 1:2, c(1, 2), byrow = TRUE)
set_left_border(ht, where(ht == 1), 1)
```

left_border_color	<i>Border colors</i>
-------------------	----------------------

Description

Functions to get or set the border colors property of huxtable table cells.

Usage

```
left_border_color(ht)
left_border_color(ht) <- value
set_left_border_color(ht, row, col, value, byrow = FALSE)

right_border_color(ht)
right_border_color(ht) <- value
set_right_border_color(ht, row, col, value, byrow = FALSE)

top_border_color(ht)
top_border_color(ht) <- value
set_top_border_color(ht, row, col, value, byrow = FALSE)

bottom_border_color(ht)
bottom_border_color(ht) <- value
set_bottom_border_color(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A vector or matrix of colors. Set to NA for the default. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Details

Both LaTeX and HTML collapse borders. If results are not what you expected, try setting the adjoining border of the previous cell to width 0 (e.g. for a left border color, unset the right border of the cell on the left).

Value

For `left_border_color`, the `left_border_color` attribute. For `set_left_border_color`, the `ht` object.

Similarly for the other functions.

See Also

[set_all_border_colors\(\)](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
left_border_color(ht) <- 'red'
left_border_color(ht)
```

```
ht <- huxtable(a = 1:3, b = 3:1)
set_left_border_color(ht, 'red')
set_left_border_color(ht, 1:2, 1, 'red')
set_left_border_color(ht, 1:2, 1:2, c('red', 'blue'), byrow = TRUE)
set_left_border_color(ht, where(ht == 1), 'red')
```

left_padding

Cell padding

Description

Functions to get or set the cell padding property of huxtable table cells.

Usage

```
left_padding(ht)
left_padding(ht) <- value
set_left_padding(ht, row, col, value, byrow = FALSE)
```

```
right_padding(ht)
right_padding(ht) <- value
set_right_padding(ht, row, col, value, byrow = FALSE)
```

```
bottom_padding(ht)
bottom_padding(ht) <- value
set_bottom_padding(ht, row, col, value, byrow = FALSE)
```

```
top_padding(ht)
top_padding(ht) <- value
set_top_padding(ht, row, col, value, byrow = FALSE)
```

```
set_all_padding(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
value	A vector or matrix. Characters must be valid CSS or LaTeX lengths. Numbers will be interpreted as lengths in points. Set to NA to reset to the default.
byrow	If TRUE, fill in values by row rather than by column.

Details

set_all_padding is a convenience function which sets left, right, top and bottom cell padding for the specified cells.

Value

For left_padding, the left_padding attribute. For set_left_padding, the ht object. Similarly for the other functions.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
left_padding(ht) <- 20
left_padding(ht)

ht <- huxtable(a = 1:3, b = 3:1)
ht2 <- set_left_padding(ht, 20)
left_padding(ht2)
ht3 <- set_left_padding(ht, 1:2, 1, 20)
left_padding(ht3)
ht4 <- set_left_padding(ht, 1:2, 1:2, c(20, 10), byrow = TRUE)
left_padding(ht4)
ht5 <- set_left_padding(ht, where(ht == 1), 20)
left_padding(ht5)
ht <- huxtable(a = 1:3, b = 1:3)
ht <- set_all_padding(ht, 1:3, 1:2, "20px")
left_padding(ht)
right_padding(ht)
```

mutate.huxtable

Dplyr verbs for huxtable

Description

Huxtable can be used with dplyr verbs [dplyr::select\(\)](#), [dplyr::rename\(\)](#), [dplyr::slice\(\)](#), [dplyr::arrange\(\)](#), [dplyr::mutate\(\)](#) and [dplyr::transmute\(\)](#). These will return huxtables. Other verbs like [dplyr::summarize\(\)](#) will simply return data frames as normal; [dplyr::pull\(\)](#) will return a vector. mutate has an extra option, detailed below.

Usage

```
## S3 method for class 'huxtable'
mutate(.data, ..., copy_cell_props = TRUE)
```

Arguments

```
.data          A huxtable.
...            Arguments passed to dplyr::mutate().
copy_cell_props
               Logical: copy cell and column properties from existing columns.
```

Details

If `mutate` creates new columns, and the argument `copy_cell_props` is missing or `TRUE`, then cell and column properties will be copied from existing columns to their left, if there are any. Otherwise, they will be the standard defaults. Row and table properties, and properties of cells in existing columns, remain unchanged.

Examples

```
ht <- hux(a = 1:5, b = 1:5, c = 1:5, d = 1:5)
bold(ht)[c(1, 3), ] <- TRUE
bold(ht)[, 1] <- TRUE
ht2 <- dplyr::select(ht, b:c)
ht2
bold(ht2)
ht3 <- dplyr::mutate(ht, x = a + b)
ht3
bold(ht3)
ht4 <- dplyr::mutate(ht, x = a + b, copy_cell_props = FALSE)
bold(ht4)
```

na_string

NA string

Description

Functions to get or set the na string property of huxtable table cells.

Usage

```
na_string(ht)
na_string(ht) <- value
set_na_string(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A character string. This will be used to replace NA values in the display. Set to NA for the default, which is the empty string. To get literal "NA", set to "NA". Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Value

For `na_string`, the `na_string` attribute. For `set_na_string`, the `ht` object.

See Also

Other formatting functions: [background_color](#), [bold](#), [font_size](#), [font](#), [number_format](#), [text_color](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
na_string(ht) <- '--'
na_string(ht)
ht[2,2] <- NA
print_screen(ht)

ht <- huxtable(a = 1:3, b = 3:1)
ht2 <- set_na_string(ht, '--')
na_string(ht2)
ht3 <- set_na_string(ht, 1:2, 1, '--')
na_string(ht3)
ht4 <- set_na_string(ht, 1:2, 1:2, c('--', ''), byrow = TRUE)
na_string(ht4)
ht5 <- set_na_string(ht, where(ht == 1), '--')
na_string(ht5)
```

number_format

Number format

Description

Functions to get or set the number format property of huxtable table cells.

Usage

```
number_format(ht)
number_format(ht) <- value
set_number_format(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A vector or list which may be character, numeric or function. See below. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Details

Number formatting is applied to any parts of cells that look like numbers (defined as an optional minus sign, followed by numerals, followed by an optional decimal point and further numerals). The exception is exponents in scientific notation; huxtable attempts to detect and ignore these.

If value is numeric, numbers will be rounded to that many decimal places. If value is character, it will be taken as an argument to [sprintf\(\)](#). If value is a function it will be applied to the numbers and should return a string. If value is NA, then numbers will be unchanged. Note that setting value to NA does not reset to the default.

The default value is "%.3g" which rounds numbers if they have more than 3 significant digits, and which may use an exponent for large numbers.

To set number_format to a function, enclose the function in list. See the examples.

Versions of huxtable before 2.0.0 applied number_format only to cells that looked like numbers in their entirety. The default value was "%5.2f".

Value

For number_format, the number_format attribute. For set_number_format, the ht object.

See Also

Other formatting functions: [background_color](#), [bold](#), [font_size](#), [font](#), [na_string](#), [text_color](#)

Examples

```
ht <- huxtable(a = 10^(3:6) + (5 * 10^(-2:-5)), b = 10^(3:6) + (5* 10^(-2:-5)))
number_format(ht)[1,] <- 2
number_format(ht)[2,] <- '%5.2f'
number_format(ht)[3,] <- list(function(x) prettyNum(x, big.mark = ','))
number_format(ht)[4,] <- list(function(x) if(x>0) '+' else '-')
ht
print_screen(ht)
ht_bands <- huxtable("10000 Maniacs")
ht_bands # probably not what you want
number_format(ht_bands) <- NA
ht_bands

ht <- huxtable(a = 1:3, b = 3:1)
set_number_format(ht, 2)
```

```
set_number_format(ht, 1:2, 1, 2)
set_number_format(ht, 1:2, 1:2, c(2, 3), byrow = TRUE)
set_number_format(ht, where(ht == 1), 2)
```

position	<i>Table position</i>
----------	-----------------------

Description

Functions to get or set the table-level table position property of a huxtable.

Usage

```
position(ht)
position(ht) <- value
set_position(ht, value)
```

Arguments

ht	A huxtable.
value	A length-one character vector which may be 'left', 'center', 'right' or NA.

Details

If your tables are too far to the right under LaTeX, try setting their `width()` explicitly.

Value

For `position`, the position attribute. For `set_position`, the ht object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
position(ht) <- 'right'
position(ht)
```

print.huxtable	<i>Default print method for huxtables</i>
----------------	---

Description

By default huxtables are printed using `print_screen()`. In certain cases, for example in Sweave documents, it may be useful to change this. You can do so by setting `options(huxtable.print)`.

Usage

```
## S3 method for class 'huxtable'  
print(x, ...)
```

```
## S3 method for class 'huxtable'  
format(x, ..., output)
```

Arguments

x	A huxtable.
...	Options passed to other methods.
output	One of "html", "latex", "md" or "screen"

Value

`print` prints the huxtable and returns NULL invisibly.

`format` returns a string representation from `to_latex()`, `to_html()` etc.

Examples

```
## Not run:  
# to print LaTeX output:  
options(huxtable.print = print_latex)  
# to print huxtables like data frames:  
options(huxtable.print = function(x, ...) print(as.data.frame(x)))  
  
## End(Not run)  
ht <- hux(a = 1:3, b = 4:6)  
format(ht, output = 'screen')  
format(ht, output = 'md')
```

print_html	<i>Create HTML representing a huxtable</i>
------------	--

Description

These functions print or return an HTML table.

Usage

```
print_html(ht, ...)  
to_html(ht, ...)  
print_notebook(ht, ...)  
  
## S3 method for class 'huxtable'  
to_html(ht, ...)
```

Arguments

ht	A huxtable.
...	Arguments to pass to methods. Not currently used.

Value

to_html returns an HTML string. print_html prints the string and returns NULL.
print_notebook prints HTML output suitable for use in an RStudio interactive notebook.

See Also

Other printing functions: [print_latex](#), [print_md](#), [print_screen](#)

Examples

```
ht <- hux(a = 1:3, b = letters[1:3])  
to_html(ht)
```

print_latex *Create LaTeX representing a huxtable*

Description

Create LaTeX representing a huxtable

Usage

```
print_latex(ht, ...)  
  
to_latex(ht, ...)  
  
## S3 method for class 'huxtable'  
to_latex(ht, tabular_only = FALSE, ...)
```

Arguments

ht A huxtable.
... Arguments to pass to methods.
tabular_only Return only the LaTeX tabular, not the surrounding float.

Value

to_latex returns a string. print_latex prints the string and returns NULL.

See Also

Other printing functions: [print_html](#), [print_md](#), [print_screen](#)

Examples

```
ht <- huxtable(a = 1:3, b = letters[1:3])  
print_latex(ht)
```

print_md *Create Markdown representing a huxtable*

Description

Create Markdown representing a huxtable

Usage

```

print_md(ht, ...)

to_md(ht, ...)

## S3 method for class 'huxtable'
to_md(ht, header = TRUE,
      min_width = getOption("width")/4, max_width = 80, ...)

```

Arguments

ht	A huxtable.
...	Arguments passed to methods.
header	Logical. Print the first row as a header?
min_width	Minimum width in on-screen characters of the result.
max_width	Maximum width in on-screen characters of the result. Overrides min_width.

Details

Only align and caption properties are used. The markdown format is `multiline_tables`, see the [rmarkdown documentation](#).

Value

`to_md` returns a string. `print_md` prints the string and returns `NULL`.

See Also

Other printing functions: [print_html](#), [print_latex](#), [print_screen](#)

Examples

```

ht <- huxtable(a = 1:5, b = 1:5)
print_md(ht)

```

```
print_screen
```

```
Print a huxtable on screen
```

Description

Print a huxtable on screen

Usage

```
print_screen(ht, ...)

to_screen(ht, ...)

## S3 method for class 'huxtable'
to_screen(ht, min_width = ceiling(getOption("width")/6),
  max_width = getOption("width", Inf), compact = TRUE, colnames = TRUE,
  color = getOption("huxtable.color_screen", default = TRUE), ...)
```

Arguments

ht	A huxtable.
...	Passed on to to_screen.
min_width	Minimum width in on-screen characters of the result.
max_width	Maximum width in on-screen characters of the result. Overrides min_width.
compact	Logical. To save space, don't print lines for empty horizontal borders.
colnames	Logical. Whether or not to print column names.
color	Logical. Whether to print the huxtable in color (requires the crayon package).

Details

colspan, rowspan, align and caption properties are shown. If the crayon package is installed, output will be colorized (and contents bolded or italicized) by default; this will work in recent daily builds of RStudio as of October 2017.

Value

to_screen returns a string. print_screen prints the string and returns NULL.

See Also

Other printing functions: [print_html](#), [print_latex](#), [print_md](#)

Examples

```
ht <- huxtable(a = 1:5, b = 1:5, add_colnames = TRUE)
ht <- set_all_borders(ht, 1:6, 1:2, 1)
right_border(ht)[,1] <- left_border(ht)[,2] <- 0
align(ht)[1,] <- 'left'
print_screen(ht)
```

`quick-output`*Quickly print objects to a PDF, HTML, Word or Excel document.*

Description

These functions use `huxtable` to print objects to an output document. They are useful as one-liners for data reporting.

Usage

```
quick_pdf(..., file = confirm("huxtable-output.pdf"), borders = 0.4,  
          open = interactive())
```

```
quick_html(..., file = confirm("huxtable-output.html"), borders = 0.4,  
          open = interactive())
```

```
quick_docx(..., file = confirm("huxtable-output.docx"), borders = 0.4,  
          open = interactive())
```

```
quick_xlsx(..., file = confirm("huxtable-output.xlsx"), borders = 0.4,  
          open = interactive())
```

Arguments

<code>...</code>	One or more <code>huxtables</code> or R objects with an <code>as_huxtable</code> method.
<code>file</code>	File path for the output.
<code>borders</code>	Border width for members of <code>...</code> that are not <code>huxtables</code> .
<code>open</code>	Logical. Automatically open the resulting file?

Details

Objects in `...` will be converted to `huxtables`, with borders added.

If `'file'` is not specified, the command will fail in non-interactive sessions. In interactive sessions, the default file path is `"huxtable-output.xxx"` in the working directory; if this already exists, you will be asked to confirm manually before proceeding.

Value

Invisible `NULL`.

Examples

```
## Not run:  
m <- matrix(1:4, 2, 2)  
dfr <- data.frame(a = 1:5, b = 1:5)  
quick_pdf(m, dfr)  
quick_html(m, dfr)
```

```
quick_docx(m, dfr)
quick_xlsx(m, dfr)

## End(Not run)
```

```
report_latex_dependencies
      Report LaTeX dependencies
```

Description

Prints out and returns a list of LaTeX dependencies for adding to a LaTeX preamble.

Usage

```
report_latex_dependencies(quiet = FALSE, as_string = FALSE)
```

Arguments

`quiet` Logical: suppress printing.
`as_string` Logical: return dependencies as a string.

Value

If `as_string` is TRUE, a string of `"\usepackage{...}"` statements; otherwise a list of `rmarkdown::latex_dependency` objects, invisibly.

Examples

```
report_latex_dependencies()
```

```
rotation      Text rotation
```

Description

Functions to get or set the text rotation property of huxtable table cells.

Usage

```
rotation(ht)
rotation(ht) <- value
set_rotation(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A numeric vector. Clockwise from the x axis, so 0 is left to right, 90 is going up, etc. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Details

You will probably need to set [col_width\(\)](#) and [row_height\(\)](#) explicitly to achieve a nice result, in both HTML and LaTeX.

Value

For rotation, the rotation attribute. For `set_rotation`, the ht object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
rotation(ht) <- 90
rotation(ht)

ht <- huxtable(a = 1:3, b = 3:1)
ht2 <- set_rotation(ht, 90)
rotation(ht2)
ht3 <- set_rotation(ht, 1:2, 1, 90)
rotation(ht3)
ht4 <- set_rotation(ht, 1:2, 1:2, c(90, 270), byrow = TRUE)
rotation(ht4)
ht5 <- set_rotation(ht, where(ht == 1), 90)
rotation(ht5)
```

rowspan	<i>Row span</i>
---------	-----------------

Description

Functions to get or set the row span property of huxtable table cells.

Usage

```
rowspan(ht)
rowspan(ht) <- value
set_rowspan(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	An integer vector or matrix of integers. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Value

For `rowspan`, the `rowspan` attribute. For `set_rowspan`, the `ht` object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
rowspan(ht)[1, 1] <- 2
rowspan(ht)
print_screen(ht)
```

rowspecs

Row and column specifications

Description

This help page describes how to use the `row` and `col` arguments in `set_*` functions.

The basics

The `set_*` functions for cell properties all have arguments like this: `set_property(ht, row, col, value, byrow = FALSE)`

You can treat `row` and `col` arguments like arguments for [data frame subsetting](#). For example, you can use `row = 1:3` to get the first three rows, `col = "salary"` to specify the column named "salary", or `row = ht$salary >= 50000` to specify rows where a condition is true.

There are also a few extra tricks you can use:

- Write `set_property(ht, x)`, omitting `row` and `col`, to set the property to `x` for all cells.
- Use `everywhere` to refer to all rows or all columns.
- Use `final(n)` to refer to the last `n` rows or columns.
- Use `evens` to get only even rows/columns and `odds` for only odd ones.
- Use `every(n, from = m)` to get every `n`th row/column starting at row/column `m`.
- Use `dplyr` functions like `starts_with`, `contains` and `matches` to specify columns (but not rows). See [select_helpers](#) for a full list.
- Use `where(condition)`, and omit the `col` argument, to get cells where `condition` is TRUE.
- Set `byrow = TRUE` to set properties by row rather than by column.

[`=`[.data.frame]: R:=[.data.frame

The gory details

How the row and col arguments are parsed depends on the number of arguments passed to the `set_*` function.

- If there are two arguments (excluding `byrow`) then the second argument is taken as the value and is set for all rows and columns.
- If there are three arguments, then the third argument is taken as the value, and `row` must be a matrix with two columns. Each row of this matrix gives the row, column indices of a single cell. This uses R's little known feature of subsetting with matrices - see [base::Extract\(\)](#).
- If there are four arguments:
 - If `row` or `col` is numeric, character or logical, it is evaluated just as in standard subsetting. `col` will be evaluated in a special context provided by `tidyselect::with_vars()` to allow the use of dplyr functions.
 - If `row` or `col` is a function, it is called with two arguments: the huxtable, and the dimension number being evaluated, i.e. 1 for rows, 2 for columns. It must return a vector of column indices. `evens()`, `odds()`, `every()` and `final()` return functions for this purpose.

Examples

```
ht <- huxtable(a = 1:5, b = 5:1)

set_bold(ht, 2:4, 1:2, TRUE)
set_bold(ht, odds, evens, TRUE)
set_bold(ht, everywhere, tidyselect::matches('[aeiou]'), TRUE)

set_bold(ht, where(ht == 1), TRUE)

set_text_color(ht, 2:3, 1:2, c('red', 'blue'))
set_text_color(ht, 2:3, 1:2, c('red', 'blue'), byrow = TRUE)
```

row_height	<i>Row heights</i>
------------	--------------------

Description

Functions to get or set the row heights property of huxtable rows.

Usage

```
row_height(ht)
row_height(ht) <- value
set_row_height(ht, row, value)
```

Arguments

<code>ht</code>	A huxtable.
<code>value</code>	A vector.
<code>row</code>	A row specifier. See rowspecs for details.

Details

If character, value must contain valid CSS or LaTeX lengths. If numeric, in HTML, values are scaled to 1 and treated as proportions of the table height. In LaTeX, they are treated as proportions of the text height (`\textheight`).

Value

For `row_height`, the `row_height` attribute. For `set_row_height`, the `ht` object.

See Also

Other row/column heights: [col_width](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
row_height(ht) <- c(.2, .1, .1)
row_height(ht)
```

sanitize

Sanitize table elements

Description

This is copied over from `xtable::sanitize()`.

Usage

```
sanitize(str, type = "latex")
```

Arguments

<code>str</code>	A character object.
<code>type</code>	"latex" or "html".

Value

The sanitized character object.

Examples

```
foo <- 'Make $$$ with us'
sanitize(foo, type = 'latex')
```

set_all_borders	<i>Set all borders</i>
-----------------	------------------------

Description

This is a convenience function which sets left, right, top and bottom borders for the specified cells.

Usage

```
set_all_borders(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
value	A numeric vector or matrix giving border widths in points. Set to 0 for no border. Set to NA to reset to the default.
byrow	If TRUE, fill in values by row rather than by column.

Value

The modified huxtable.

See Also

[left_border\(\)](#), [set_outer_borders\(\)](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
set_all_borders(ht, 1:3, 1:2, 1)
```

set_all_border_colors	<i>Set all border colors</i>
-----------------------	------------------------------

Description

This is a convenience function which sets left, right, top and bottom border colors for the specified cells.

Usage

```
set_all_border_colors(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
value	A vector or matrix of colors. Set to NA for the default. Set to NA to reset to the default.
byrow	If TRUE, fill in values by row rather than by column.

Value

The modified huxtable.

See Also

[left_border_color\(\)](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
ht <- set_all_border_colors(ht, 'red')
```

set_cell_properties *Set multiple cell properties*

Description

Set multiple cell properties

Usage

```
set_cell_properties(ht, row, col, ...)
```

Arguments

ht	A huxtable.
row	A row specification.
col	A column specification.
...	Named list of property values.

Value

The modified huxtable object.

Examples

```
ht <- hux(a = 1:3, b = 1:3)
ht <- set_cell_properties(ht, 1, 1, italic = TRUE, text_color = 'red')
text_color(ht)
ht
```

set_default_properties

Set default huxtable properties

Description

Defaults are used for new huxtables, and also when a property is set to NA.

Usage

```
set_default_properties(...)
```

Arguments

... Properties specified by name, or a single named list.

Value

A list of the previous property values, invisibly.

See Also

[get_default_properties\(\)](#)

Examples

```
old <- set_default_properties(left_border = 1)
hux(a = 1:2, b = 1:2)
set_default_properties(old)
```

set_outer_borders *Set outer borders round a rectangle of cells*

Description

This is a convenience function to set a border round the top, bottom, left and right of a group of cells.

Usage

```
set_outer_borders(ht, row, col, value)
```

Arguments

ht	A huxtable
row	A row specifier. See rowspecs() for details.
col	A column specifier.
value	A numeric value for the border width. Set to 0 for no border.

Value

The modified huxtable.

See Also

[left_border\(\)](#), [set_all_borders\(\)](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
set_outer_borders(ht, 1)
set_outer_borders(ht, 2:3, 1:2, 1)
```

t.huxtable *Transpose a huxtable*

Description

Transpose a huxtable

Usage

```
## S3 method for class 'huxtable'
t(x)
```

Arguments

x A huxtable.

Details

Row and column spans of x will be swapped, as will column widths and row heights, table width and height, and cell borders (bottom becomes right, etc.). Other properties - in particular, alignment, vertical alignment and rotation - will be preserved.

Value

The transposed object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
bottom_border(ht)[3,] <- 1
t(ht)
```

tabular_environment *Tabular environment*

Description

Functions to get or set the table-level tabular environment property of a huxtable.

Usage

```
tabular_environment(ht)
tabular_environment(ht) <- value
set_tabular_environment(ht, value)
```

Arguments

ht A huxtable.
value A length-one character vector. Set to NA for the default, 'tabularx'.

Details

No features are guaranteed to work if you set this to a non-default value. Use at your own risk!

Value

For tabular_environment, the tabular_environment attribute. For set_tabular_environment, the ht object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
tabular_environment(ht) <- 'longtable'
tabular_environment(ht)
```

text_color	<i>Text color</i>
------------	-------------------

Description

Functions to get or set the text color property of huxtable table cells.

Usage

```
text_color(ht)
text_color(ht) <- value
set_text_color(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A character vector or matrix of valid R color names. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Value

For text_color, the text_color attribute. For set_text_color, the ht object.

See Also

Other formatting functions: [background_color](#), [bold](#), [font_size](#), [font](#), [na_string](#), [number_format](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
text_color(ht) <- 'blue'
text_color(ht)
```

```
ht <- huxtable(a = 1:3, b = 3:1)
set_text_color(ht, 'blue')
```



```

set_text_color(ht, 1:2, 1, 'blue')
set_text_color(ht, 1:2, 1:2, c('blue', 'red'), byrow = TRUE)
set_text_color(ht, where(ht == 1), 'blue')

```

 themes

Theme a huxtable

Description

These functions quickly set default styles for a huxtable.

Usage

```

theme_plain(ht, position = "left")

theme_basic(ht, header_row = TRUE, header_col = TRUE)

theme_stripped(ht, stripe = grDevices::grey(0.9), header_row = TRUE,
  header_col = TRUE)

theme_article(ht, header_row = TRUE, header_col = TRUE)

```

Arguments

ht	A huxtable object.
position	'left', 'centre' or 'right'
header_row	Logical: style first row differently?
header_col	Logical: style first column differently?
stripe	Background colour for alternate rows

Details

theme_plain is a simple theme with a bold header, a grey striped background, and an outer border.

theme_basic just adds a border for header rows and/or columns.

theme_stripped uses different backgrounds for alternate rows, and for headers.

theme_article is similar to the style of many scientific journals. It sets horizontal lines above and below the table.

Value

The huxtable object, appropriately styled.

Examples

```

ht <- huxtable(a = 1:5, b = 1:5)
theme_stripped(ht)

```

valign	<i>Vertical alignment</i>
--------	---------------------------

Description

Functions to get or set the vertical alignment property of huxtable table cells.

Usage

```
valign(ht)
valign(ht) <- value
set_valign(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A character vector or matrix which may be 'top', 'middle', 'bottom' or NA. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Details

Vertical alignment may not work for short text in LaTeX. Defining row heights with [row_height\(\)](#) may help.

Value

For valign, the valign attribute. For set_valign, the ht object.

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
valign(ht) <- 'bottom'
valign(ht)

ht <- huxtable(a = 1:3, b = 3:1)
ht2 <- set_valign(ht, 'bottom')
valign(ht2)
ht3 <- set_valign(ht, 1:2, 1, 'bottom')
valign(ht3)
ht4 <- set_valign(ht, 1:2, 1:2, c('bottom', 'bottom'), byrow = TRUE)
valign(ht4)
ht5 <- set_valign(ht, where(ht == 1), 'bottom')
valign(ht5)
```

where	<i>Return array indices where expression is true</i>
-------	--

Description

This is a simple wrapper around `which(..., arr.ind = TRUE)`, for use in [row specifications](#).

Usage

```
where(expr)
```

Arguments

expr	An R expression
------	-----------------

Value

A matrix of row and column indices of cells where `expr` is TRUE.

Examples

```
ht <- hux(a = 1:3, b = 4:6, add_colnames = TRUE)
where(ht > 2)
where(is_a_number(ht))
```

width	<i>Table width</i>
-------	--------------------

Description

Functions to get or set the table-level table width property of a huxtable.

Usage

```
width(ht)
width(ht) <- value
set_width(ht, value)
```

Arguments

ht	A huxtable.
value	A length-one vector. If numeric, value is treated as a proportion of the surrounding block width (HTML) or text width (LaTeX). If character, it must be a valid CSS or LaTeX width. Set to NA for the default.

Value

For width, the width attribute. For set_width, the ht object.

See Also

Other table measurements: [height](#)

Examples

```
ht <- huxtable(a = 1:3, b = 1:3)
width(ht) <- 0.8
width(ht)
```

wrap

Text wrapping

Description

Functions to get or set the text wrapping property of huxtable table cells.

Usage

```
wrap(ht)
wrap(ht) <- value
set_wrap(ht, row, col, value, byrow = FALSE)
```

Arguments

ht	A huxtable.
value	A logical vector or matrix. If TRUE, long cell contents will be wrapped into multiple lines. Set to NA for the default. Set to NA to reset to the default.
row	A row specifier. See rowspecs for details.
col	An optional column specifier.
byrow	If TRUE, fill in values by row rather than by column.

Value

For wrap, the wrap attribute. For set_wrap, the ht object.

Examples

```

ht <- huxtable(a = rep('Some long text', 2))
wrap(ht)[1,] <- TRUE
print_html(ht)

ht <- huxtable(a = 1:3, b = 3:1)
ht2 <- set_wrap(ht, TRUE)
wrap(ht2)
ht3 <- set_wrap(ht, 1:2, 1, TRUE)
wrap(ht3)
ht4 <- set_wrap(ht, 1:2, 1:2, c(TRUE, FALSE), byrow = TRUE)
wrap(ht4)
ht5 <- set_wrap(ht, where(ht == 1), TRUE)
wrap(ht5)

```

[.huxtable]

*Subset a huxtable***Description**

Subset a huxtable

Usage

```

## S3 method for class 'huxtable'
x[i, j, drop = FALSE]

## S3 replacement method for class 'huxtable'
x[i, j] <- value

## S3 replacement method for class 'huxtable'
x$name <- value

## S3 replacement method for class 'huxtable'
x[[i, j]] <- value

```

Arguments

x	A huxtable.
i	Rows to select.
j, name	Columns to select.
drop	Not used.
value	A matrix, data frame, huxtable or similar object.

Details

[always returns a new huxtable object, while \$ and [[simply return a vector of data. For the replacement function [<, if value is a huxtable, then its cell properties will be copied into x. In addition, if value fills up an entire column, then column properties will be copied into the replaced columns of x, and if it fills up an entire row, then row properties will be copied into the replaced rows of x. Replacement functions \$<- and [[<- simply change the data without affecting other properties.

Value

A huxtable.

Examples

```
ht <- huxtable(a = 1:3, b = letters[1:3])
ht[1:2,]
ht[,1]
ht$a
## Not run:
rowspan(ht)[2,1] <- 2
ht[1:2,] # generates a warning

## End(Not run)
ht <- huxtable(a = 1:3, b = 1:3)
ht2 <- huxtable(10:11, 12:13)
bold(ht2) <- TRUE
ht[2:3,] <- ht2
ht
bold(ht)
```

Index

`[.huxtable`, 61
`[<-.huxtable` (`[.huxtable`), 61
`[[<-.huxtable` (`[.huxtable`), 61
`$<-.huxtable` (`[.huxtable`), 61

`add_colnames`, 4
`add_footnote`, 5
`add_rownames` (`add_colnames`), 4
`align`, 6
`align()`, 25
`align<-` (`align`), 6
`as_FlexTable`, 7
`as_flextable` (`as_FlexTable`), 7
`as_hux` (`huxtable`), 24
`as_huxtable` (`huxtable`), 24
`as_huxtable()`, 3, 4
`as_Workbook`, 8

`background_color`, 9, 10, 19, 20, 37, 38, 56
`background_color<-` (`background_color`), 9
`base::Extract()`, 49
`bold`, 9, 10, 19, 20, 37, 38, 56
`bold<-` (`bold`), 10
`bottom_border` (`left_border`), 31
`bottom_border<-` (`left_border`), 31
`bottom_border_color`
 (`left_border_color`), 33
`bottom_border_color<-`
 (`left_border_color`), 33
`bottom_padding` (`left_padding`), 34
`bottom_padding<-` (`left_padding`), 34
`broom::glance()`, 24
`broom::tidy()`, 23, 24

`caption`, 11
`caption()`, 12
`caption<-` (`caption`), 11
`caption_pos`, 12
`caption_pos()`, 11
`caption_pos<-` (`caption_pos`), 12

`cbind.data.frame()`, 13
`cbind.huxtable`, 13
`cbind.huxtable()`, 27
`col_width`, 15, 50
`col_width()`, 47
`col_width<-` (`col_width`), 15
`colspan`, 14
`colspan<-` (`colspan`), 14

`data frame subsetting`, 48
`data.frame()`, 25
`dplyr-verbs` (`mutate.huxtable`), 35
`dplyr::arrange()`, 35
`dplyr::mutate()`, 35, 36
`dplyr::pull()`, 35
`dplyr::rename()`, 35
`dplyr::select()`, 35
`dplyr::slice()`, 35
`dplyr::summarize()`, 35
`dplyr::transmute()`, 35

`escape_contents`, 16
`escape_contents<-` (`escape_contents`), 16
`evens` (`every`), 17
`evens()`, 49
`every`, 17
`every()`, 49
`everywhere` (`every`), 17

`filter`, 18
`final`, 18
`final()`, 49
`flextable::flextable()`, 7
`font`, 9, 10, 19, 20, 37, 38, 56
`font<-` (`font`), 19
`font_size`, 9, 10, 19, 20, 37, 38, 56
`font_size<-` (`font_size`), 20
`format.huxtable` (`print.huxtable`), 40

`get_default_properties`, 21

- get_default_properties(), 53
- glue::glue(), 24
- guess_knitr_output_format, 21
- height, 22, 60
- height(), 8
- height<- (height), 22
- hux (huxtable), 24
- hux_logo, 26
- huxreg, 23
- huxtable, 24
- huxtable(), 3, 4
- huxtable-package, 3, 26
- insert_column, 27
- insert_row (insert_column), 27
- is_a_number, 28
- is_a_number(), 8
- is_hux (huxtable), 24
- is_huxtable (huxtable), 24
- italic (bold), 10
- italic<- (bold), 10
- italic<-<- (bold), 10
- knit_print.data.frame, 28, 30
- knit_print.huxtable, 29, 29
- knitr::knit_print(), 30
- label, 30
- label<- (label), 30
- latex_float, 31
- latex_float<- (latex_float), 31
- left_border, 31
- left_border(), 51, 54
- left_border<- (left_border), 31
- left_border_color, 33
- left_border_color(), 52
- left_border_color<- (left_border_color), 33
- left_padding, 34
- left_padding<- (left_padding), 34
- mutate (mutate.huxtable), 35
- mutate.huxtable, 35
- mutate.huxtable(), 18
- na_string, 9, 10, 19, 20, 36, 38, 56
- na_string<- (na_string), 36
- number_format, 9, 10, 19, 20, 37, 37, 56
- number_format(), 23, 25
- number_format<- (number_format), 37
- odds (every), 17
- odds(), 49
- officer::read_docx(), 7
- openxlsx::openxlsx(), 8
- openxlsx::saveWorkbook(), 8
- pad_decimal (align), 6
- pad_decimal<- (align), 6
- position, 39
- position(), 12
- position<- (position), 39
- print.huxtable, 40
- print.huxtable(), 3
- print_html, 41, 42–44
- print_latex, 41, 42, 43, 44
- print_md, 41, 42, 42, 44
- print_notebook (print_html), 41
- print_screen, 41–43, 43
- print_screen(), 40
- quick-output, 45
- quick_docx (quick-output), 45
- quick_html (quick-output), 45
- quick_pdf (quick-output), 45
- quick_xlsx (quick-output), 45
- rbind.huxtable (cbind.huxtable), 13
- report_latex_dependencies, 46
- right_border (left_border), 31
- right_border<- (left_border), 31
- right_border_color (left_border_color), 33
- right_border_color<- (left_border_color), 33
- right_padding (left_padding), 34
- right_padding<- (left_padding), 34
- rotation, 46
- rotation<- (rotation), 46
- row specifications, 59
- row_height, 15, 49
- row_height(), 47, 58
- row_height<- (row_height), 49
- rowspan, 47
- rowspan<- (rowspan), 47
- rowspecs, 6, 9, 10, 14–20, 32, 33, 35, 37, 38, 47, 48, 48, 49, 51, 52, 56, 58, 60
- rowspecs(), 54

- sanitize, 50
- select_helpers, 48
- set_align (align), 6
- set_all_border_colors, 51
- set_all_border_colors(), 34
- set_all_borders, 51
- set_all_borders(), 32, 54
- set_all_padding (left_padding), 34
- set_background_color
(background_color), 9
- set_bold (bold), 10
- set_bottom_border (left_border), 31
- set_bottom_border_color
(left_border_color), 33
- set_bottom_padding (left_padding), 34
- set_caption (caption), 11
- set_caption_pos (caption_pos), 12
- set_cell_properties, 52
- set_cell_properties(), 5
- set_col_width (col_width), 15
- set_colspan (colspan), 14
- set_default_properties, 53
- set_default_properties(), 21
- set_escape_contents (escape_contents),
16
- set_font (font), 19
- set_font_size (font_size), 20
- set_height (height), 22
- set_italic (bold), 10
- set_italic<- (bold), 10
- set_label (label), 30
- set_latex_float (latex_float), 31
- set_left_border (left_border), 31
- set_left_border_color
(left_border_color), 33
- set_left_padding (left_padding), 34
- set_na_string (na_string), 36
- set_number_format (number_format), 37
- set_outer_borders, 54
- set_outer_borders(), 51
- set_pad_decimal (align), 6
- set_position (position), 39
- set_right_border (left_border), 31
- set_right_border_color
(left_border_color), 33
- set_right_padding (left_padding), 34
- set_rotation (rotation), 46
- set_row_height (row_height), 49
- set_rowspan (rowspan), 47
- set_set_italic (bold), 10
- set_tabular_environment
(tabular_environment), 55
- set_text_color (text_color), 56
- set_top_border (left_border), 31
- set_top_border_color
(left_border_color), 33
- set_top_padding (left_padding), 34
- set_valign (valign), 58
- set_width (width), 59
- set_wrap (wrap), 60
- sprintf(), 38
- t.huxtable, 54
- tabular_environment, 55
- tabular_environment<-
(tabular_environment), 55
- text_color, 9, 10, 19, 20, 37, 38, 56
- text_color<- (text_color), 56
- theme_article (themes), 57
- theme_basic (themes), 57
- theme_plain (themes), 57
- theme_plain(), 3, 29
- theme_stripped (themes), 57
- themes, 57
- tidyselect::with_vars(), 49
- to_html (print_html), 41
- to_html(), 40
- to_latex (print_latex), 42
- to_latex(), 40
- to_md (print_md), 42
- to_screen (print_screen), 43
- top_border (left_border), 31
- top_border<- (left_border), 31
- top_border_color (left_border_color), 33
- top_border_color<- (left_border_color),
33
- top_padding (left_padding), 34
- top_padding<- (left_padding), 34
- valign, 58
- valign<- (valign), 58
- where, 59
- where(condition), 48
- width, 22, 59
- width(), 8, 39
- width<- (width), 59

wrap, [60](#)

wrap<- (wrap), [60](#)