Visual Test of typstable

Contents

`tt()` Base

Default with rownames=TRUE

Default table includes row names as the first column.

```
tt(mt)
```

	mpg	cyl	disp
Mazda RX4	21	6	160
Mazda RX4 Wag	21	6	160
Datsun 710	22.8	4	108
Hornet 4 Drive	21.4	6	258
Hornet Sportabout	18.7	8	360

rownames=FALSE

Exclude row names entirely.

```
tt(mt, rownames = FALSE)
```

mpg	cyl	disp
21	6	160

21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

cols with tidy-select subset

Select specific columns with tidy-select syntax.

```
tt(mtcars[1:5, ], cols = c(mpg, cyl, hp), rownames = FALSE)
```

mpg	cyl	hp
21	6	110
21	6	110
22.8	4	93
21.4	6	110
18.7	8	175

col_names custom names

Override display names for columns.

```
tt(mt, col_names = c("", "Miles/Gallon", "Cylinders", "Displacement"))
```

	Miles/Gallon	Cylinders	Displacement
Mazda RX4	21	6	160
Mazda RX4 Wag	21	6	160
Datsun 710	22.8	4	108
Hornet 4 Drive	21.4	6	258
Hornet Sportabout	18.7	8	360

align single value

Apply a single alignment to all columns.

```
tt(mt, align = "center", rownames = FALSE)
```

mpg	cyl	disp

21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

align per-column vector

Specify alignment for each column individually.

```r
tt(mt, align = c("left", "right", "center"), rownames = FALSE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

col_widths explicit

Provide explicit column widths via the `col_widths` parameter.

```r
tt(mt, col_widths = c("100pt", "2fr", "1fr"), rownames = FALSE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

escape=TRUE with special characters

Typst special characters are escaped by default.

```r
special <- data.frame(
  col1 = c("*bold*", "_underline_", "#heading"),
  col2 = c("@ref", "<angle>", "a > b"),
  stringsAsFactors = FALSE
)
tt(special, escape = TRUE, rownames = FALSE)
```

col1	col2
bold	@ref
underline	<angle>
#heading	a > b

escape=FALSE with raw Typst markup

Allow raw Typst markup to pass through unescaped.

```r
raw_typst <- data.frame(
  col1 = c("#strong[Bold]", "#emph[Italic]", "plain"),
  col2 = c("a", "b", "c"),
  stringsAsFactors = FALSE
)
tt(raw_typst, escape = FALSE, rownames = FALSE)
```

col1	col2
Bold	a
Italic	b
plain	c

Quarto cross-reference

Test Quarto `tbl-cap` and `label` options.

```r
tt(mt, rownames = FALSE) |>
  tt_style(stroke = TRUE)
```

Table 1: This is a caption

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

See Table 1.

`tt_widths()`

Positional proportional widths

Set widths as proportional values applied in column order.

```r
tt(mt, rownames = FALSE) |>
  tt_widths(1, 3, 1)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Named column widths

Target specific columns by name.

```r
tt(mt, rownames = FALSE) |>
  tt_widths(mpg = 2, cyl = 1, disp = 3)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258

18.7	8	360

Equal explicit widths

All columns get equal proportions.

```r
tt(mt, rownames = FALSE) |>
  tt_widths(1, 1, 1)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

`tt_style()`

stroke=TRUE (full grid)

Add default borders to all cells.

```r
tt(mt, rownames = FALSE) |>
  tt_style(stroke = TRUE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

stroke with color

Add blue 2pt borders.

```r
tt(mt, rownames = FALSE) |>
  tt_style(stroke = "2pt + blue")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

striped=TRUE

Alternate row fill colors.

```
tt(mt, rownames = FALSE) |>
  tt_style(striped = TRUE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

fill (global)

Set a fill color for the entire table.

```
tt(mt, rownames = FALSE) |>
  tt_style(fill = "#f5f5dc")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

inset (global padding)

Set cell padding globally.

```
tt(mt, rownames = FALSE) |>
  tt_style(inset = "8pt")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

row_gutter with stroke

Add vertical spacing between rows.

```
tt(mt, rownames = FALSE) |>
  tt_style(stroke = TRUE, row_gutter = "4pt")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

column_gutter with stroke

Add horizontal spacing between columns.

```
tt(mt, rownames = FALSE) |>
  tt_style(stroke = TRUE, column_gutter = "4pt")
```

mpg	cyl	disp
21	6	160
21	6	160

22.8	4	108
21.4	6	258
18.7	8	360

position="center" with full_width=FALSE

Center a non-full-width table on the page.

```
tt(mt, rownames = FALSE) |>
  tt_style(position = "center", full_width = FALSE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

full_width=TRUE

Force the table to span the full page width.

```
tt(mt, rownames = FALSE) |>
  tt_style(full_width = TRUE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

`tt_column()` Static Styling

bold=TRUE

Make the mpg column bold.

```r
tt(mt, rownames = FALSE) |>
  tt_column(mpg, bold = TRUE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

italic=TRUE

Make the cyl column italic.

```r
tt(mt, rownames = FALSE) |>
  tt_column(cyl, italic = TRUE)
```

mpg	*cyl*	disp
21	*6*	160
21	*6*	160
22.8	*4*	108
21.4	*6*	258
18.7	*8*	360

color (static string)

Apply blue text color to a column.

```r
tt(mt, rownames = FALSE) |>
  tt_column(mpg, color = "blue")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

fill (static)

Apply yellow fill to a column.

```r
tt(mt, rownames = FALSE) |>
  tt_column(disp, fill = "yellow")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

font_size

Set a smaller font size on a column.

```r
tt(mt, rownames = FALSE) |>
  tt_column(mpg, font_size = "8pt")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

rotate (numeric)

Rotate column content by 90 degrees using a numeric value.

```r
tt(mt, rownames = FALSE) |>
  tt_column(cyl, rotate = 90)
```

mpg	cyl	disp
21	6	160
21	6	160

22.8	4	108
21.4	6	258
18.7	8	360

rotate (string)

Rotate column content by 45 degrees using a string value.

```r
tt(mt, rownames = FALSE) |>
  tt_column(cyl, rotate = "45deg")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

inset

Apply custom cell padding to a column.

```r
tt(mt, rownames = FALSE) |>
  tt_column(mpg, inset = "12pt")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

width override

Override the width of a single column.

```
tt(mt, rownames = FALSE) |>
  tt_column(mpg, width = "100pt")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

border_left=TRUE

Add a left border to a column.

```
tt(mt, rownames = FALSE) |>
  tt_column(cyl, border_left = TRUE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

border_right with stroke spec

Add a styled right border to a column.

```
tt(mt, rownames = FALSE) |>
  tt_column(cyl, border_right = "2pt + red")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108

21.4	6	258
18.7	8	360

`tt_column()` Tidy-Select

Multiple columns with c()

Style multiple columns at once.

```
tt(mt, rownames = FALSE) |>
  tt_column(c(mpg, disp), bold = TRUE, color = "navy")
```

mpg	cyl	**disp**
21	6	**160**
21	6	**160**
22.8	4	**108**
21.4	6	**258**
18.7	8	**360**

everything() selector

Apply styling to all columns.

```
tt(mt, rownames = FALSE) |>
  tt_column(everything(), fill = "#f0f0f0")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Numeric index

Target a column by its numeric position.

```
tt(mt, rownames = FALSE) |>
  tt_column(1, italic = TRUE, color = "red")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

`tt_column()` Data-Driven

fill via bare column reference

Use a hidden column to drive fill colors per row.

```
df_bg <- data.frame(
  name = c("Alice", "Bob", "Carol"),
  score = c(95, 72, 88),
  bg = c("green", "red", "yellow"),
  stringsAsFactors = FALSE
)
tt(df_bg, cols = c(name, score), rownames = FALSE) |>
  tt_column(score, fill = bg)
```

name	score
Alice	95
Bob	72
Carol	88

color via pattern

Use `{col}` pattern to find per-column color columns.

```
df_color <- data.frame(
  a = c(10, 20, 30),
  b = c(40, 50, 60),
  color_a = c("red", "green", "blue"),
  color_b = c("purple", "orange", "black"),
  stringsAsFactors = FALSE
)
tt(df_color, cols = c(a, b), rownames = FALSE) |>
  tt_column(c(a, b), color = "color_{col}")
```

a	b
10	40
20	50
30	60

bold via bare column reference

Use a boolean column to conditionally bold cells.

```r
df_bold <- data.frame(
  item = c("A", "B", "C"),
  value = c(1, 2, 3),
  is_bold = c(TRUE, FALSE, TRUE),
  stringsAsFactors = FALSE
)
tt(df_bold, cols = c(item, value), rownames = FALSE) |>
  tt_column(value, bold = is_bold)
```

item	value
A	**1**
B	2
C	**3**

italic via pattern

Use `{col}` pattern for per-column italic flags.

```r
df_italic <- data.frame(
  x = c("foo", "bar", "baz"),
  italic_x = c(TRUE, FALSE, TRUE),
  stringsAsFactors = FALSE
)
tt(df_italic, cols = x, rownames = FALSE) |>
  tt_column(x, italic = "italic_{col}")
```

x
foo
bar
baz

font_size via pattern

Drive font size from a data column.

```
df_size <- data.frame(
  label = c("Small", "Medium", "Large"),
  size_label = c("8pt", "10pt", "14pt"),
  stringsAsFactors = FALSE
)
tt(df_size, cols = label, rownames = FALSE) |>
  tt_column(label, font_size = "size_{col}")
```

label
Small
Medium
Large

rotate via bare column reference

Drive rotation angle from a data column.

```
df_rot <- data.frame(
  text = c("Up", "Tilt", "Down"),
  angle = c("90deg", "45deg", "-45deg"),
  stringsAsFactors = FALSE
)
tt(df_rot, cols = text, rownames = FALSE) |>
  tt_column(text, rotate = angle)
```

text
Up
Tilt
Down

.missing="ignore" with nonexistent pattern column

Silently skip when a pattern column does not exist.

```
tt(mt, rownames = FALSE) |>
  tt_column(c(mpg, cyl), color = "color_{col}", .missing = "ignore")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

inset via pattern

Drive cell padding from a data column.

```r
df_inset <- data.frame(
  val = c(1, 2, 3),
  inset_val = c("4pt", "8pt", "16pt"),
  stringsAsFactors = FALSE
)
tt(df_inset, cols = val, rownames = FALSE) |>
  tt_column(val, inset = "inset_{col}")
```

val
1
2
3

`tt_row()`

Header row styling

Style the header row (row 0) with bold, fill, and color.

```r
tt(mt, rownames = FALSE) |>
  tt_row(0, bold = TRUE, fill = "navy", color = "white")
```

mpg	**cyl**	**disp**
21	6	160
21	6	160
22.8	4	108

21.4	6	258
18.7	8	360

Single data row fill and color

Highlight a single data row.

```
tt(mt, rownames = FALSE) |>
  tt_row(2, fill = "#ffffcc", color = "darkred")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Multiple rows with fill

Style rows 1, 3, and 5 simultaneously.

```
tt(mt, rownames = FALSE) |>
  tt_row(c(1, 3, 5), fill = "#e0e0ff")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

italic and font_size on a row

Apply italic and custom font size to a row.

```
tt(mt, rownames = FALSE) |>
  tt_row(1, italic = TRUE, font_size = "8pt")
```

mpg	cyl	disp

21	*6*	*160*
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

align override on a row

Override alignment for a specific row.

```r
tt(mt, rownames = FALSE) |>
  tt_row(2, align = "center")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

rotate on a data row

Rotate all content in a data row.

```r
tt(mt, rownames = FALSE) |>
  tt_row(1, rotate = 45)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

inset on a row

Apply extra padding to a row.

```r
tt(mt, rownames = FALSE) |>
  tt_row(3, inset = "10pt")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

hline_above and hline_below

Add horizontal lines around a row.

```r
tt(mt, rownames = FALSE) |>
  tt_row(3, hline_above = TRUE, hline_below = "2pt + red")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

tt_cell()

fill and bold on a cell

Highlight a specific cell with fill and bold.

```r
tt(mt, rownames = FALSE) |>
  tt_cell(1, 1, fill = "yellow", bold = TRUE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108

21.4	6	258
18.7	8	360

italic, color, and font_size on a cell

Apply multiple styles to a single cell.

```
tt(mt, rownames = FALSE) |>
  tt_cell(2, 2, italic = TRUE, color = "red", font_size = "14pt")
```

mpg	cyl	disp
21	6	160
21	*6*	160
22.8	4	108
21.4	6	258
18.7	8	360

align override on a cell

Override alignment for one cell.

```
tt(mt, rownames = FALSE) |>
  tt_cell(1, 1, align = "right")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

rotate on a cell

Rotate a single cell.

```
tt(mt, rownames = FALSE) |>
  tt_cell(3, 1, rotate = 90)
```

mpg	cyl	disp

21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

inset on a cell

Add extra padding to a specific cell.

```
tt(mt, rownames = FALSE) |>
  tt_cell(2, 3, inset = "12pt")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

colspan=2 with content override

Span a cell across two columns with replacement content.

```
tt(mt, rownames = FALSE) |>
  tt_cell(1, 1, colspan = 2, content = "Combined")
```

mpg	cyl	disp
Combined		6
160	21	6
160	22.8	4
108	21.4	6
258	18.7	8
360		

rowspan=3 with content override

Span a cell across three rows with replacement content.

```r
tt(mt, rownames = FALSE) |>
  tt_cell(1, 1, rowspan = 3, content = "Grouped")
```

mpg	cyl	disp
Grouped	6	160
	21	6
	160	22.8
4	108	21.4
6	258	18.7
8	360	

content override with styling

Replace cell content and apply formatting.

```r
tt(mt, rownames = FALSE) |>
  tt_cell(1, 1, content = "CUSTOM", bold = TRUE, color = "blue", fill =
"#f0f0f0")
```

mpg	cyl	disp
CUSTOM	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Header cell styling

Style a cell in the header row (row 0).

```r
tt(mt, rownames = FALSE) |>
  tt_cell(0, 1, bold = TRUE, color = "red", fill = "lightyellow")
```

mpg	cyl	disp
21	6	160

21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

tt_hline() and tt_vline()

Custom hline at position 3

Add a horizontal line below the third data row.

```
tt(mt, rownames = FALSE) |>
  tt_hline(3)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Hline with styled stroke

Add a blue 2pt horizontal line.

```
tt(mt, rownames = FALSE) |>
  tt_hline(2, stroke = "2pt + blue")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Partial hline with start/end

A horizontal line spanning only some columns.

```r
tt(mt, rownames = FALSE) |>
  tt_hline(2, start = 0, end = 2)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Vline after column 1

Add a vertical line after the first column.

```r
tt(mt, rownames = FALSE) |>
  tt_vline(1)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Vline with styled stroke

Add a gray vertical line.

```r
tt(mt, rownames = FALSE) |>
  tt_vline(1, stroke = "1pt + gray")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Partial vline with start/end

A vertical line spanning only some rows.

```r
tt(mt, rownames = FALSE) |>
  tt_vline(1, start = 1, end = 5)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

`tt_pack_rows()`

Single group with default indent and bold

Pack rows into one labeled group.

```r
tt(mt10, rownames = FALSE) |>
  tt_pack_rows("Group A", 1, 5)
```

mpg	cyl	disp
Group A		
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360
18.1	6	225
14.3	8	360
24.4	4	146.7
22.8	4	140.8
19.2	6	167.6

Multiple chained groups

Chain multiple pack_rows calls.

```r
tt(mt10, rownames = FALSE) |>
  tt_pack_rows("Group A", 1, 5) |>
  tt_pack_rows("Group B", 6, 10)
```

mpg	cyl	disp
Group A		
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360
Group B		
18.1	6	225
14.3	8	360
24.4	4	146.7
22.8	4	140.8
19.2	6	167.6

index parameter

Define multiple groups with a named vector.

```r
tt(mt10, rownames = FALSE) |>
  tt_pack_rows(index = c("First Five" = 5, "Last Five" = 5))
```

mpg	cyl	disp
First Five		
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Last Five		
18.1	6	225
14.3	8	360
24.4	4	146.7
22.8	4	140.8
19.2	6	167.6

indent=FALSE

Disable indentation of grouped rows.

```
tt(mt10, rownames = FALSE) |>
  tt_pack_rows("No Indent Group", 1, 5, indent = FALSE)
```

mpg	cyl	disp
No Indent Group		
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360
18.1	6	225
14.3	8	360
24.4	4	146.7
22.8	4	140.8
19.2	6	167.6

bold_label=FALSE

Use a non-bold group label.

```
tt(mt10, rownames = FALSE) |>
  tt_pack_rows("Light Label", 1, 5, bold_label = FALSE)
```

mpg	cyl	disp
Light Label		

21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360
18.1	6	225
14.3	8	360
24.4	4	146.7
22.8	4	140.8
19.2	6	167.6

Basic `tt_header_above()`

Single spanning header

Add one spanning header group above the columns.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("All Columns" = 6))
```

All Columns					
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Multiple groups with default gap

Group columns into multiple labeled spans.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2))
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Groups with empty spacer

Use a space label for ungrouped columns.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c(" " = 1, "Performance" = 2, "Characteristics" = 3))
```

	Performance		Characteristics		
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

gap=NULL (no gap columns)

Disable gap columns between header groups.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2), gap =
NULL)
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

gap="20pt" (wider gap)

Use wider gap columns between header groups.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2), gap =
"20pt")
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Custom styling (bold=FALSE, color, fill, align)

Override the default bold and apply colors.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(
    c("Performance" = 2, "Engine" = 2, "Design" = 2),
    bold = FALSE,
    color = "white",
    fill = "navy",
    align = "left"
  )
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

line=FALSE

Disable the horizontal line below the spanning header.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2), line =
FALSE)
```

Performance		**Engine**		**Design**	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

italic + font_size

Use the `italic` and `font_size` parameters.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(
    c("Performance" = 2, "Engine" = 2, "Design" = 2),
    italic = TRUE, font_size = "14pt"
  )
```

Performance		***Engine***		***Design***	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

rotate

Rotate the header_above text.

```
tt(mt4, rownames = FALSE) |>
  tt_header_above(c("AB" = 2, "CD" = 2), rotate = 45, gap = NULL)
```

AB		CD	
mpg	cyl	disp	hp
21	6	160	110
21	6	160	110
22.8	4	108	93
21.4	6	258	110
18.7	8	360	175

inset

Custom padding on the header_above row.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(
    c("Performance" = 2, "Engine" = 2, "Design" = 2),
    inset = "12pt", fill = "#f0f0f0"
  )
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Stacked Headers

Two stacked header rows (no gaps)

Add two levels of spanning headers without gaps.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2), gap =
NULL) |>
  tt_header_above(c("Car Metrics" = 6))
```

Car Metrics					
Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Three stacked headers with different styling (no gaps)

Multiple header levels with distinct visual styles.

```
tt(mt6, rownames = FALSE, align = 'center') |>
  tt_header_above(c("A" = 2, "B" = 2, "C" = 2)) |>
  tt_header_above(c("Left" = 3, "Right" = 3), fill = "lightgrey") |>
  tt_header_above(c("All" = 6), fill = "navy", color = "white")
```

All					
Left			Right		
A		B		C	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Two stacked headers with gaps on innermost

Stacked headers where the innermost header has gaps. Outer header colspans auto-adjust.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_header_above(c("Car Metrics" = 6))
```

Car Metrics					
Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Three stacked headers with gaps

Three stacked levels where the innermost has gaps. Outer headers adjust to accommodate gap columns.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("A" = 2, "B" = 2, "C" = 2)) |>
  tt_header_above(c("Left" = 3, "Right" = 3), fill = "lightgray") |>
  tt_header_above(c("All" = 6), fill = "navy", color = "white")
```

All					
Left			**Right**		
A		**B**		**C**	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Targeting Headers with `tt_row()` (Negative Indices)

Use `tt_row()` with negative row indices to style `tt_header_above()` rows. Row −1 is the innermost (closest to the main header), −2 is the next up, etc.

Italic via tt_row(-1)

Override the innermost header_above row with italic.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_row(-1, italic = TRUE)
```

Performance		***Engine***		***Design***	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32

Performance		***Engine***		***Design***	
mpg	qsec	cyl	disp	hp	wt
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Fill via tt_row(-1)

Apply a fill color to the innermost header_above row.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_row(-1, fill = "#cce5ff")
```

Performance		**Engine**		**Design**	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Override bold=FALSE via tt_row(-1)

The header_above defaults to bold=TRUE. Override it with `tt_row()`.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_row(-1, bold = FALSE, color = "navy")
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Color + font_size via tt_row(-1)

Combine color and font_size overrides.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_row(-1, color = "red", font_size = "14pt")
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Style different stacked rows independently

Target each stacked header_above row by its negative index.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("A" = 2, "B" = 2, "C" = 2)) |>
  tt_header_above(c("All" = 6)) |>
  tt_row(-1, fill = "#ffffcc", italic = TRUE) |>
  tt_row(-2, fill = "navy", color = "white")
```

All					
A		*B*		*C*	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875

All					
A		***B***		***C***	
mpg	qsec	cyl	disp	hp	wt
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Three stacked rows with individual styles

Each header level gets a distinct visual treatment.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("A" = 2, "B" = 2, "C" = 2), gap = NULL) |>
  tt_header_above(c("Left" = 3, "Right" = 3), gap = NULL) |>
  tt_header_above(c("Title" = 6)) |>
  tt_row(-1, fill = "#e8f5e9") |>
  tt_row(-2, fill = "#e3f2fd") |>
  tt_row(-3, fill = "#fce4ec", italic = TRUE)
```

Title					
Left			**Right**		
A		**B**		**C**	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Targeting Headers with `tt_cell()` (Negative Indices)

Use `tt_cell()` with negative row indices to style individual header groups. The column is automatically normalized to the start of the group it falls within.

Fill on one group

Apply fill to just the “Engine” group (cols 3-4).

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_cell(-1, 3, fill = "lightyellow")
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Color on one group

Apply red text to the “Design” group (cols 5-6).

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_cell(-1, 5, color = "red")
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Content override on a group

Replace the label of a header_above group.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_cell(-1, 3, content = "Motor")
```

Performance		Motor		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Column normalization: any column in group hits the same cell

Targeting column 1 or column 2 both hit the “Performance” group.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_cell(-1, 2, fill = "lightcoral", italic = TRUE)
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Style multiple groups independently

Give each header group a distinct appearance.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_cell(-1, 1, fill = "#cce5ff") |>
  tt_cell(-1, 3, fill = "#d4edda") |>
  tt_cell(-1, 5, fill = "#f8d7da")
```

Performance		**Engine**		**Design**	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

tt_cell on outer stacked header

Target a group in an outer (non-innermost) header_above row.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("A" = 2, "B" = 2, "C" = 2), gap = NULL) |>
  tt_header_above(c("Left" = 3, "Right" = 3), gap = NULL) |>
  tt_cell(-2, 4, fill = "lightyellow", italic = TRUE)
```

Left			***Right***		
A		**B**		**C**	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Content + styling override on stacked header

Replace label and style a group in a stacked configuration.

```r
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("A" = 2, "B" = 2, "C" = 2), gap = NULL) |>
  tt_header_above(c("Summary" = 6)) |>
  tt_cell(-2, 1, content = "Overview", color = "navy", bold = FALSE) |>
  tt_cell(-1, 3, fill = "#fff3cd")
```

Overview					
A		**B**		**C**	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Combined: tt_row + tt_cell on Header Above

Row-level style + cell-level override

`tt_row(-1)` sets a base style; `tt_cell(-1, ...)` overrides for one group.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Design" = 2)) |>
  tt_row(-1, fill = "#e0e0e0", italic = TRUE) |>
  tt_cell(-1, 3, fill = "lightyellow", italic = FALSE)
```

Performance		**Engine**		***Design***	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

header_spec defaults + tt_row + tt_cell precedence

The header_spec provides base defaults (bold=TRUE, align=center). `tt_row` overrides at row level, `tt_cell` overrides at cell level. Later calls win.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(
    c("Performance" = 2, "Engine" = 2, "Design" = 2),
    color = "gray"
  ) |>
  tt_row(-1, color = "blue") |>
  tt_cell(-1, 5, color = "red", bold = FALSE)
```

Performance		**Engine**		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Full example: stacked headers with mixed targeting

Comprehensive example combining header_above params, tt_row, and tt_cell.

```
tt(mt6, rownames = FALSE) |>
  tt_header_above(
    c("Performance" = 2, "Engine" = 2, "Design" = 2),
    italic = TRUE
  ) |>
  tt_header_above(
    c("Car Data" = 6),
    fill = "navy", color = "white"
  ) |>
  tt_row(-1, fill = "#f0f0f0") |>
  tt_cell(-1, 1, fill = "#cce5ff") |>
  tt_cell(-1, 3, fill = "#d4edda") |>
  tt_cell(-1, 5, fill = "#f8d7da") |>
  tt_cell(-2, 1, content = "All Car Metrics")
```

All Car Metrics					
Performance		***Engine***		***Design***	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875

All Car Metrics					
Performance		***Engine***		***Design***	
mpg	qsec	cyl	disp	hp	wt
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46

Precedence / Last-Write-Wins

Row after column (row wins)

When row styling is applied after column styling, the row's fill wins on conflicting attributes.

```r
tt(mt, rownames = FALSE) |>
  tt_column(mpg, fill = "lightblue") |>
  tt_row(2, fill = "lightyellow")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Column after row (column wins)

When column styling is applied after row styling, the column's fill wins on conflicting attributes.

```r
tt(mt, rownames = FALSE) |>
  tt_row(2, fill = "lightyellow") |>
  tt_column(mpg, fill = "lightblue")
```

mpg	cyl	disp
21	6	160
21	6	160

22.8	4	108
21.4	6	258
18.7	8	360

Cell overrides both row and column

Cell-level styling always takes highest precedence when applied last.

```
tt(mt, rownames = FALSE) |>
  tt_column(mpg, fill = "lightblue") |>
  tt_row(2, fill = "lightyellow") |>
  tt_cell(2, 1, fill = "lightgreen")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Non-conflicting attributes merge

Different attributes from column, row, and cell merge together.

```
tt(mt, rownames = FALSE) |>
  tt_column(mpg, bold = TRUE) |>
  tt_row(2, fill = "lightyellow") |>
  tt_cell(2, 1, italic = TRUE)
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Data-driven column after row

Data-driven column formatting applied after row styling.

```
df_prec <- data.frame(
  val = c(10, 20, 30, 40, 50),
  bg_val = c("lightcoral", "lightgreen", "lightblue", "lightyellow",
"lightpink"),
  stringsAsFactors = FALSE
)
tt(df_prec, cols = val, rownames = FALSE) |>
  tt_row(2, fill = "white") |>
  tt_column(val, fill = bg_val)
```

val
10
20
30
40
50

NA Handling

NA values in data cells

Table with NA values in the data.

```
df_na <- data.frame(
  a = c(1, NA, 3),
  b = c(NA, "hello", NA),
  c = c(TRUE, FALSE, NA),
  stringsAsFactors = FALSE
)
tt(df_na, rownames = FALSE)
```

a	b	c
1		TRUE
	hello	FALSE
3		

NA in data-driven style columns

NA values in columns used for data-driven styling.

```
df_na_style <- data.frame(
  val = c(10, 20, 30),
  bg = c("lightblue", NA, "lightyellow"),
  stringsAsFactors = FALSE
)
tt(df_na_style, cols = val, rownames = FALSE) |>
  tt_column(val, fill = bg)
```

val
10
20
30

Complex Compositions

Combined alignment with rotation

Use combined alignment like `"left + bottom"` with rotation.

```
tt(mt, rownames = FALSE) |>
  tt_row(0, rotate = -45, align = "left + bottom")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

Full-featured table

Combine header_above, pack_rows, hlines, vlines, column, row, and cell styles.

```
tt(mtcars[1:10, 1:6], rownames = FALSE) |>
  tt_header_above(c("Performance" = 2, "Engine" = 2, "Other" = 2)) |>
  tt_pack_rows("Group 1", 1, 5) |>
  tt_pack_rows("Group 2", 6, 10) |>
  tt_hline(1) |>
  tt_vline(2, stroke = "1pt + gray") |>
  tt_column(mpg, bold = TRUE, color = "navy") |>
  tt_row(0, fill = "#e0e0e0") |>
  tt_cell(1, 1, fill = "lightyellow") |>
  tt_style(stroke = TRUE, inset = "4pt")
```

Performance		Engine		Other	
mpg	cyl	disp	hp	drat	wt
Group 1					
21	6	160	110	3.9	2.62
21	6	160	110	3.9	2.875
22.8	4	108	93	3.85	2.32
21.4	6	258	110	3.08	3.215
18.7	8	360	175	3.15	3.44
Group 2					
18.1	6	225	105	2.76	3.46
14.3	8	360	245	3.21	3.57
24.4	4	146.7	62	3.69	3.19
22.8	4	140.8	95	3.92	3.15
19.2	6	167.6	123	3.92	3.44

pack_rows and header_above combined

Use both row grouping and spanning headers.

```
tt(mt10, rownames = FALSE) |>
  tt_header_above(c("A" = 1, "B" = 2)) |>
  tt_pack_rows("First", 1, 5) |>
  tt_pack_rows("Second", 6, 10) |>
  tt_style(stroke = TRUE)
```

A	B	
mpg	cyl	disp
First		
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360
Second		

A	B	
mpg	cyl	disp
18.1	6	225
14.3	8	360
24.4	4	146.7
22.8	4	140.8
19.2	6	167.6

Multiple tt_column calls on same column (last wins)

The last call's conflicting attributes take precedence.

```r
tt(mt, rownames = FALSE) |>
  tt_column(mpg, color = "red", bold = TRUE) |>
  tt_column(mpg, color = "blue")
```

mpg	cyl	disp
21	6	160
21	6	160
22.8	4	108
21.4	6	258
18.7	8	360

colspan and rowspan in same table

Use both column spanning and row spanning cells.

```r
df_span <- data.frame(
  a = c("r1", "r2", "r3", "r4"),
  b = c("x", "y", "z", "w"),
  c = c(1, 2, 3, 4),
  d = c(5, 6, 7, 8),
  stringsAsFactors = FALSE
)
tt(df_span, rownames = FALSE) |>
  tt_cell(1, 1, rowspan = 2, content = "Rows 1-2") |>
  tt_cell(3, 2, colspan = 2, content = "Cols 2-3")
```

a	b	c	d

Rows 1-2	x	1	5
	r2	y	2
6	r3	Cols 2-3	
3	7	r4	w
4	8		

Color formats

Test different color specification formats.

```r
df_colors <- data.frame(
  hex = c("Hex color", "Another hex", "Third hex"),
  named = c("Named Typst", "Also named", "Yet another"),
  rcolor = c("R color", "Another R", "Third R"),
  stringsAsFactors = FALSE
)
tt(df_colors, rownames = FALSE) |>
  tt_column(hex, color = "#ff5733") |>
  tt_column(named, color = "eastern") |>
  tt_column(rcolor, color = "darkgreen") |>
  tt_cell(1, 1, fill = "#e0f0ff") |>
  tt_cell(2, 2, fill = "aqua") |>
  tt_cell(3, 3, fill = "lavender")
```

hex	named	rcolor
Hex color	Named Typst	R color
Another hex	Also named	Another R
Third hex	Yet another	Third R

Custom pattern

```r
tt(mt6,rownames=FALSE) |>
  tt_header_above(c("Performance"=2,"Engine"=2,"Design"=2)) |>
  tt_cell(row=1,column=3, fill='pat')
```

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
21	16.46	6	160	110	2.62
21	17.02	6	160	110	2.875

Performance		Engine		Design	
mpg	qsec	cyl	disp	hp	wt
22.8	18.61	4	108	93	2.32
21.4	19.44	6	258	110	3.215
18.7	17.02	8	360	175	3.44
18.1	20.22	6	225	105	3.46