

## rowtable

Edit tables row-by row. The idea is a row-oriented way to input tables, with just a little less syntactical overhead than the usual `table` function in Typst.

The function does a shallow split of the input row by a configurable separator, which is `&` by default.

Input: `[A & B & C]`

Table cells (effectively): `..([A], [B], [C])`

For improved table ergonomics, the table sizes the number of columns by the longest row. All rows are effectively completed so that they are of full length. This creates a better the editing experience, as rows can be filled out gradually.

## Examples

### Document Result

<i>goá</i>	<i>íáu-boē</i>	<i>koat-tēng</i>	<i>tang-sî</i>	<i>boeh</i>	<i>tńg-khì</i>
goa <sup>1</sup>	iau <sup>1</sup> -boe <sup>3</sup>	koat <sup>2</sup> -teng <sup>3</sup>	tang <sup>7</sup> -si <sup>5</sup>	boeh <sup>2</sup>	tng <sup>1</sup> -khi <sup>3</sup>
goa <sup>2</sup>	iau <sup>2</sup> -boe <sup>7</sup>	koat <sup>4</sup> -teng <sup>7</sup>	tang <sup>1</sup> -si <sup>5</sup>	boeh <sup>4</sup>	tng <sup>2</sup> -khi <sup>3</sup>
I	not-yet	decide	when	want	return.

"I have not yet decided when I shall return."

### Input

```
#{
  set table.hline(stroke: 0.08em)
  show regex("\\d"): super.with(size: 0.8em, typographic: false)
  show table.cell: it => { set text(size: 0.9em) if it.y >= 1; it }
  show table.cell.where(y: 0): emph
  rowtable(
    separator: ",", // configurable separator
    stroke: 0pt, // pass through table arguments, hlines, cells et.c.
    inset: (x: 0em),
    column-gutter: 0.9em,
    // rows are filled to be equal length after collecting cells
    [goá , íáu-boē , koat-tēng , tang-sî , boeh , tńg-khì ],
    [goa1 , iau1-boe3 , koat2-teng3 , tang7-si5 , boeh2 , tng1-khi3 ],
    [goa2 , iau2-boe7 , koat4-teng7 , tang1-si5 , boeh4 , tng2-khi3 ],
    [I , not-yet , decide , when , want , return. ],
    table.hline(),
    // cell that fills remainder of row
    expandcell["I have not yet decided when I shall return."],
  )
}
```

Example from Wikipedia<sup>1</sup>

<sup>1</sup>[https://en.wikipedia.org/wiki/Interlinear\\_gloss](https://en.wikipedia.org/wiki/Interlinear_gloss)

## Document Result





Term	Explanation	Assumptions
$X$	Explanatory variables	Non-random
$Y$	$Y_1, \dots, Y_n$ observations	<b>Pairwise independent</b>
$\beta$	Model parameters	

## Input

```
#{
  set table(stroke: none, inset: 0.8em)
  set table.hline(stroke: 0.5pt)
  show table.cell.where(y: 0): strong
  show table.cell.where(x: 0): x => math.bold(math.uptight(x))
  rowtable(
    table.hline(),
    [Term      & Explanation                & Assumptions ],
    table.hline(),
    [X$        & Explanatory variables      & Non-random ],
    [Y$        & $Y_1, ..., Y_n$ observations & *Pairwise independent*],
    [beta$     & Model parameters            ],
    table.hline(),
  )
}
```

## Trying some more difficult examples

Document Result

Literal &	Strong	X-Y
Equation $\pi = 3.1415\dots$	$\int_{\Omega} d\omega$	X&Y
<ul style="list-style-type: none"> <li>• A</li> <li>• B</li> </ul>	<ol style="list-style-type: none"> <li>1. A</li> <li>2. B</li> </ol>	<b>A</b> a <b>B</b> b
Figure 1: Top 	See Figure 1 & Figure 2	 Figure 2: Bot
Nested rowtable 	Nested table 	table.cell
Cell with colspan=2		-
Expandcell		the rest
N/A	N/A	N/A

Input

```
#rowtable(
  align: horizon,
  stroke: 0.1pt,
  row-filler: [N/A],
  [Literal \& & *Strong* & *X*--_Y_ ],
  [Equation \ $pi = 3.1415...$ & $ \int_{\Omega} d\omega $ & $X \& Y$],
  [
    - A
    - B
    &
    + A
    + B
    &
    / A: a
    / B: b
  ],
  [
    #{
      set figure.caption(position: top)
      [#figure(rect[A], caption: "Top")<fig1>]
    }
    &
    See @fig1 \& @fig2
    &
    [#figure(rect[B], caption: "Bot")<fig2>]
  ],
  {
    [Nested rowtable \ ]
    rowtable([A & B])
    [&]
    [Nested table \ ]
    table(columns: 2, [A], [B])
    [&]
    table.cell(stroke: 1pt + red)[`table.cell`]
  },
  [#table.cell(fill: yellow.lighten(90%), colspan: 2)[Cell with colspan=2] &
  #table.cell[--]],
  [#expandcell(fill: yellow.lighten(90%))[Expandcell] & #expandcell[the rest]],
  [&&],
  [],
)
```

## Double semicolon separator

*Document Result*

First	This is a literal ;; and ; and , and &
Second; Third	Equation $\pi = 3.1415\dots$

*Input*

```
#rowtable(  
  separator: ";;",  
  align: horizon,  
  stroke: (x, y) => (y: int(y <= 0) * 0.9pt + 0.1pt, x: 0.1pt),  
  [First      ;; This is a literal \;\; and ; and , and & ],  
  [Second; Third  ;; Equation  $\pi = 3.1415\dots$ ],  
  table.hline(stroke: 1pt),  
)
```