Using `auto` columns is very convenient when it is the largest cell in that column:

```
#table(
    columns: (auto, auto, auto),
    align: horizon + center,
    bdiagbox[Names][Properties], [*Can Walk*], [*Can Run*],
    [*Character A*], [Yes], [No],
    [*Character B*], [No], [No]
    )
```

Properties / Names	**Can Walk**	**Can Run**
Character A	Yes	No
Character B	No	No

Note that this fails when the diagbox is not the largest cell:

```
#table(
    columns: (auto, auto, auto),
    align: horizon + center,
    bdiagbox[Names][Properties], [*Can Walk*], [*Can Run*],
    [*Long Long Long Character A*], [Yes], [No],
    [*Long Long Long Character B*], [No], [No]
    )
```

Properties / Names	**Can Walk**	**Can Run**
Long Long Long Character A	Yes	No
Long Long Long Character B	No	No

Instead, you will have to specify sizes manually in such cases:

```
#let fst_column_size = 15em;
#table(
    columns: (fst_column_size, auto, auto),
    align: horizon + center,
    bdiagbox(width: fst_column_size)[Names][Properties], [*Can Walk*], [*Can Run*],
    [*Long Long Long Character A*], [Yes], [No],
    [*Long Long Long Character B*], [No], [No]
    )
```

Properties / Names	**Can Walk**	**Can Run**
Long Long Long Character A	Yes	No
Long Long Long Character B	No	No

You can also have diagonal lines pointing in the opposite direction, using `tdiagbox`:

```
#let third_column_size = 5em;
#table(
    columns: (auto, auto, third_column_size),
    align: horizon + center,
    bdiagbox[Names][Properties], [*Can Walk*], [*Can Run*],
    [*Character A*], [Yes], [No],
    [*Character B*], [No], tdiagbox(width: third_column_size)[A][B]
    )
```

Names \ Properties	**Can Walk**	**Can Run**
Character A	Yes	No
Character B	No	A / B

If your table has a custom `inset` (inner padding) property, make sure to pass it along:

```
#let third_column_size = 5em;
#let inset = 20pt;
#table(
    columns: (auto, auto, third_column_size),
    align: horizon + center,
    inset: inset,
    bdiagbox(inset: inset)[Names][Properties], [*Can Walk*], [*Can Run*],
    [*Character A*], [Yes], [No],
    [*Character B*], [No], tdiagbox(inset: inset, width: third_column_size)[A][B]
    )
```

Names \ Properties	**Can Walk**	**Can Run**
Character A	Yes	No
Character B	No	A / B

You may specify a standalone (table-less) diagbox with `inset: 0pt` and `box_stroke: 1pt`:

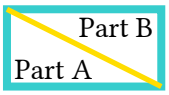
```
#bdiagbox(inset: 0pt, box_stroke: 1pt)[Part A][Part B]
```

Part A \ Part B

Additionally, diagboxes have many more properties which enable near-full customization. Most were inspired by the `diagbox` LaTeX package.

For example, you may customize the box and line colors and sizes:

```
#bdiagbox(
    inset: 0pt,
    box_stroke: teal + 3pt, line_stroke: yellow + 2pt
    )[Part A][Part B]
```

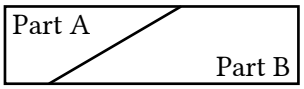


Use `left_sep` and `right_sep` to move text horizontally without resizing the box:

```
#bdiagbox(
    width: 10em,
    inset: 0pt, box_stroke: 1pt,  // standalone
    left_sep: 3em, right_sep: -0.5em
    )[Part A][Part B]
```

Use `left_outer_sep` and `right_outer_sep` to move the diagonal line horizontally (its start and its end, respectively):

```
#tdiagbox(
    width: 10em,
    inset: 0pt, box_stroke: 1pt,  // standalone
    left_outer_sep: 1.5em, right_outer_sep: 4em
    )[Part A][Part B]
```



Also note that the box's inner width (that is, total width minus inset padding) is calculated with `width - 2*inset`; you may specify the `inner_width` parameter to override that. (But you will generally want to specify just `width`, as it corresponds directly to the width of the table column the diagbox is in.)

You can also use `height` if you want to define a custom, fixed height for your box:

```
#tdiagbox(
    width: 10em, height: 5em,
    inset: 0pt, box_stroke: 1pt,  // standalone
    )[Part A][Part B]
```