# typst-ribbons

## Quick Start Guide

## 1. Introduction

typst-ribbons makes it easy to create beautiful flow diagrams in Typst. This guide will get you started in 5 minutes.
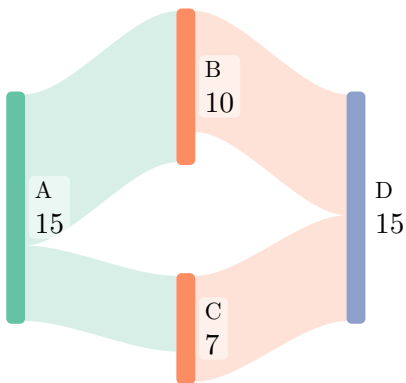
## 2. Installation

Import the library:

```
#import "src/ribbons.typ": *
```

## 3. Your First Diagram
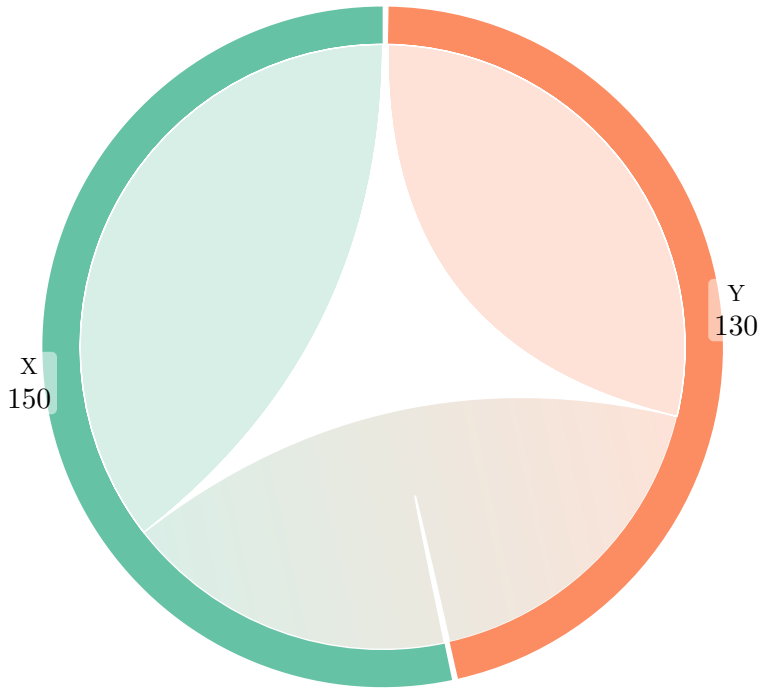
### 3.1. Simple Sankey

```
#sankey-diagram((
  "A": ("B": 10, "C": 5),
  "B": ("D": 8),
  "C": ("D": 7),
))
```



That's it! You created a Sankey diagram showing flow from A through B and C to D.

### 3.2. Simple Chord

```
#chord-diagram((
  "X": ("X": 100, "Y": 50),
  "Y": ("X": 50, "Y": 80),
))
```
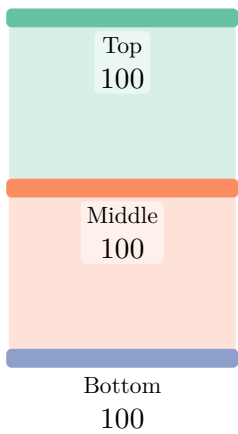
A circular chord diagram showing relationships between X and Y.

# 4. Common Customizations

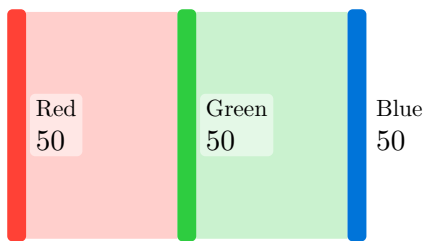### 4.1. Vertical Layout

```
#sankey-diagram(
  (
    "Top": ("Middle": 100),
    "Middle": ("Bottom": 100),
  ),
  layout: layout.auto-linear(vertical: true)
)
```
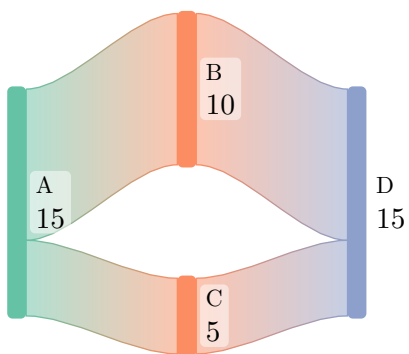
## 4.2. Custom Colors

```
#sankey-diagram(
  (
    "Red": ("Green": 50),
    "Green": ("Blue": 50),
  ),
  tinter: tinter.dict-tinter((
    "Red": red,
    "Green": green,
    "Blue": blue,
  ))
)
```



## 4.3. Prettier Ribbons
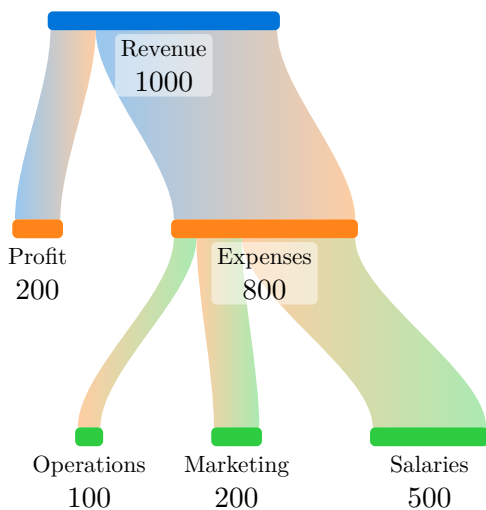
```
#sankey-diagram(
  (
    "A": ("B": 10, "C": 5),
    "B": ("D": 10),
    "C": ("D": 5),
  ),
  ribbon-stylizer: ribbon-stylizer.gradient-from-to(
    transparency: 50%,
    stroke-width: 0.5pt,
  )
)
```

# 5. Real-World Example

Here's a complete example showing a company budget:

```
#sankey-diagram(
  (
    "Revenue": ("Expenses": 800, "Profit": 200),
    "Expenses": (
      "Salaries": 500,
      "Marketing": 200,
      "Operations": 100,
    ),
  ),
  layout: layout.auto-linear(
    vertical: true,
    layer-gap: 2.5,
  ),
  tinter: tinter.layer-tinter(
    palette: (blue, orange, green)
  ),
  ribbon-stylizer: ribbon-stylizer.gradient-from-to(
    transparency: 60%,
  )
)
```

# 6. Next Steps

## 6.1. Learn More
- Read `api-reference.typ` for complete documentation
- Browse `examples.typ` for more use cases
- Check out `demo.typ` for complex examples

## 6.2. Key Parameters to Explore

**Layout:**
- `vertical: true` - Top-to-bottom flow
- `layer-gap` - Space between columns
- `node-gap` - Space between nodes

**Colors:**
- `tinter.layer-tinter()` - Color by stage
- `tinter.dict-tinter()` - Custom colors
- `palette.tableau` - Use different color schemes

**Ribbons:**
- `gradient-from-to()` - Gradient ribbons
- `transparency` - Adjust transparency (0-100%)
- `stroke-width` - Add borders

**Labels:**
- `draw-label: none` - Hide labels
- Custom label content and styling

## 6.3. Common Patterns

**Budget/Finance:**

```
layout: layout.auto-linear(vertical: true)
tinter: tinter.layer-tinter()
```

**Energy/Flow:**

```
tinter: tinter.categorical-tinter()
ribbon-stylizer: ribbon-stylizer.gradient-from-to()
```

**Relationships:**

```
chord-diagram(...)
tinter: tinter.node-tinter()
```

# 7. Tips
1. Start simple, add styling gradually
2. Use `vertical: true` for top-down processes
3. Use dict-tinter for brand colors
4. Adjust `transparency` to 50-70% for readability
5. Add `stroke-width: 0.5pt` for clearer separation

# 8. Getting Help
- Check the API reference for function details
- Look at examples.typ for similar use cases

- Experiment with parameters - most are optional!

Happy diagramming! 🎨