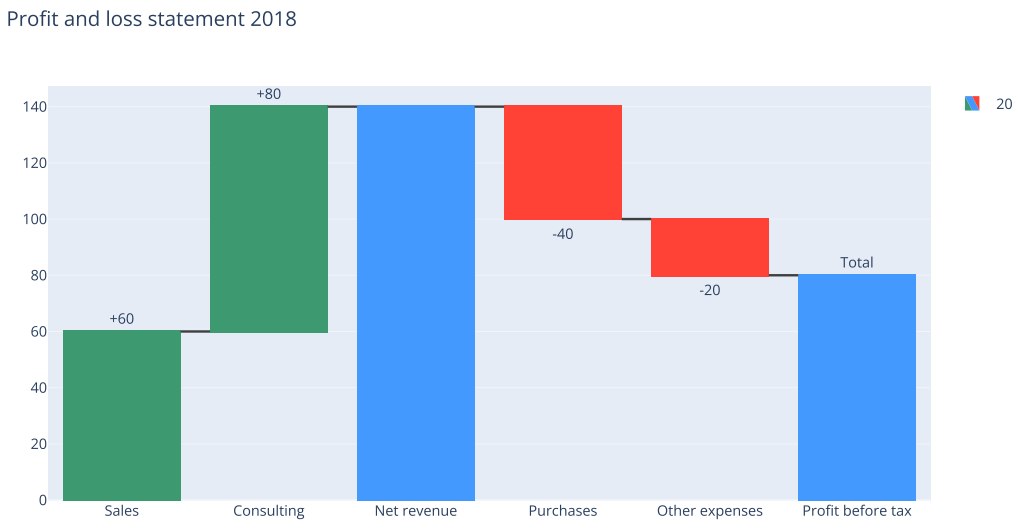
Waterfall Charts in Python

How to make waterfall plots in Python with Plotly.

> Plotly Studio: Transform any dataset into an interactive data application in minutes with AI. Sign up for early access now. (https://plotly.com/studio/?utm_medium=graphing_libraries&utm_campaign=studio_early_access&utm_content=sidebar)

Simple Waterfall Chart

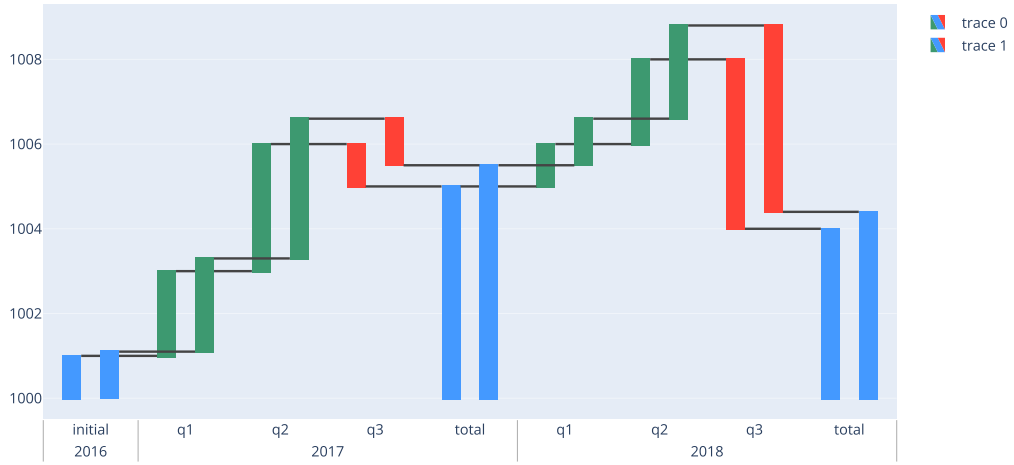
```
import plotly.graph_objects as go

fig = go.Figure(go.Waterfall(
    name = "20", orientation = "v",
    measure = ["relative", "relative", "total", "relative", "relative", "total"],
    x = ["Sales", "Consulting", "Net revenue", "Purchases", "Other expenses", "Profit before tax"],
    textposition = "outside",
    text = ["+60", "+80", "", "-40", "-20", "Total"],
    y = [60, 80, 0, -40, -20, 0],
    connector = {"line":{"color":"rgb(63, 63, 63)"}},
))

fig.update_layout(
        title = "Profit and loss statement 2018",
        showlegend = True
)

fig.show()
```



Multi-Category Waterfall Chart

[hi... aterfallgroupgap attribute (https://plotly.com/python/reference/layout/#layout-waterfallgroupgap), which sets a gap between bars.

```
import plotly.graph_objects as go

fig = go.Figure()

fig.add_trace(go.Waterfall(
    x = [["2016", "2017", "2017", "2017", "2017", "2018", "2018", "2018", "2018"],
        ["initial", "q1", "q2", "q3", "total", "q1", "q2", "q3", "total"]],
    measure = ["absolute", "relative", "relative", "relative", "total", "relative", "relative", "relative", "total"],
    y = [1, 2, 3, -1, None, 1, 2, -4, None],
    base = 1000
))

fig.add_trace(go.Waterfall(
    x = [["2016", "2017", "2017", "2017", "2017", "2018", "2018", "2018", "2018"],
        ["initial", "q1", "q2", "q3", "total", "q1", "q2", "q3", "total"]],
    measure = ["absolute", "relative", "relative", "relative", "total", "relative", "relative", "relative", "total"],
    y = [1.1, 2.2, 3.3, -1.1, None, 1.1, 2.2, -4.4, None],
    base = 1000
))

fig.update_layout(
    waterfallgroupgap = 0.5,
)

fig.show()
```



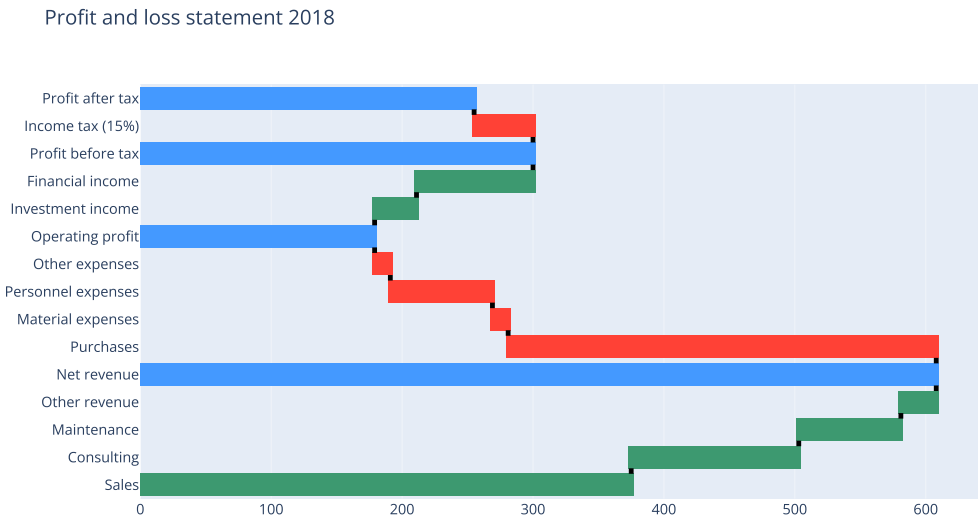
Setting Marker Size and Color

This example uses decreasing (https://plotly.com/python/reference/waterfall/#waterfall-decreasing), increasing (https://plotly.com/python/reference/waterfall/#waterfall-increasing), and totals (https://plotly.com/python/reference/waterfall/#waterfall-totals) attributes to customize the bars.

```
import plotly.graph_objects as go

fig = go.Figure(go.Waterfall(
    x = [["2016", "2017", "2017", "2017", "2017", "2018", "2018", "2018", "2018"],
       ["initial", "q1", "q2", "q3", "total", "q1", "q2", "q3", "total"]],
    measure = ["absolute", "relative", "relative", "relative", "total", "relative", "relative", "relative", "total"],
    y = [10, 20, 30, -10, None, 10, 20, -40, None], base = 300,
    decreasing = {"marker":{"color":"Maroon", "line":{"color":"red", "width":2}}},
    increasing = {"marker":{"color":"Teal"}},
    totals = {"marker":{"color":"deep sky blue", "line":{"color":"blue", "width":3}}}
))

fig.update_layout(title = "Profit and loss statement", waterfallgap = 0.3)

fig.show()
```

Horizontal Waterfall Chart

```
import plotly.graph_objects as go

fig = go.Figure(go.Waterfall(
    name = "2018", orientation = "h", measure = ["relative", "relative", "relative", "relative", "total", "relative",
                                                 "relative", "relative", "relative", "total", "relative", "relative", "total", "relative", "total"],
    y = ["Sales", "Consulting", "Maintenance", "Other revenue", "Net revenue", "Purchases", "Material expenses",
       "Personnel expenses", "Other expenses", "Operating profit", "Investment income", "Financial income",
       "Profit before tax", "Income tax (15%)", "Profit after tax"],
    x = [375, 128, 78, 27, None, -327, -12, -78, -12, None, 32, 89, None, -45, None],
    connector = {"mode":"between", "line":{"width":4, "color":"rgb(0, 0, 0)", "dash":"solid"}}
))

fig.update_layout(title = "Profit and loss statement 2018")

fig.show()
```



Reference

See https://plotly.com/python/reference/waterfall/ (https://plotly.com/python/reference/waterfall/) for more information and chart attribute options!

What About Dash?

Dash (https://dash.plot.ly/) is an open-source framework for building analytical applications, with no Javascript required, and it is tightly integrated with the Plotly graphing library.

Learn about how to install Dash at https://dash.plot.ly/installation (https://dash.plot.ly/installation).

Everywhere in this page that you see fig.show(), you can display the same figure in a Dash application by passing it to the figure argument of the Graph component (https://dash.plot.ly/dash-core-components/graph) from the built-in dash_core_components package like this:

```
import plotly.graph_objects as go # or plotly.express as px
fig = go.Figure() # or any Plotly Express function e.g. px.bar(...)
# fig.add_trace( ... )
# fig.update_layout( ... )

from dash import Dash, dcc, html

app = Dash()
app.layout = html.Div([
    dcc.Graph(figure=fig)
])

app.run(debug=True, use_reloader=False)  # Turn off reloader if inside Jupyter
```