




What is Altair?

Altair is a declarative statistical visualization library for Python and it is based on Vega-Lite.

↳ open shell and type '`pip install altair`' to install altair library.



Altair's API is simple, friendly and consistent and built on top of the powerful Vega-Lite visualization grammar. With Altair, you can spend more time understanding your data and its meaning.

The key idea of Altair is that you are declaring links between DataFrame columns and visual encoding channels, such as the x-axis, y-axis, color, etc. The rest of the plot details are handled automatically. This elegant simplicity produces beautiful and compelling visualizations with a minimal amount of code. Building on this declarative plotting idea, a surprising range of simple to sophisticated plots and visualizations can be created using a relatively simple grammar.



What is vega-lite?

- ❑ Vega-Lite is a high-level grammar of interactive graphics.
- ❑ It provides a concise JSON syntax for rapidly generating visualizations to support data analysis.
- ❑ The Vega-Lite compiler automatically produces visualization components including axes, legends, and scales.



steps mentioned below to generate an Altair chart:

- *Create an Altair Chart object with a pandas DataFrame if required.*
- *Choose a suitable mark relevant to your Dataset.*
- *Encode the X and Y values with appropriate columns in the DataFrame.*
- *Save the data emitted by Altair Chart object as a file with a .json extension.*
- *Navigate to <https://vega.github.io/editor/> an online Vega-Lite editor and paste the contents of the JSON file in the left pane of the editor. You can see a Chart generated in the right pane of the editor.*

<https://vega.github.io/editor/>

Example in next slide



Data Type supported by Altair:

Data Type	Code	Description
Quantitative	Q	Numerical Quantity (Real-Valued)
Nominal	N	Name/Unordered Categorical
Ordinal	O	Ordered Categorical
Temporal	T	Date/Time

example:

```
import pandas as pd
import altair as alt

# chart object using pandas DataFrame-
data = pd.DataFrame.from_records([
    {"a": "A", "b": 28}, {"a": "B", "b": 55}, {"a": "C", "b": 43},
    {"a": "D", "b": 91}, {"a": "E", "b": 81}, {"a": "F", "b": 53},
    {"a": "G", "b": 19}, {"a": "H", "b": 87}, {"a": "I", "b": 52}
])

# print data to see the table created of the given data-
print(data)

# create a chart using appropriate marker (bar graph in this case)-
chart=alt.Chart(data).mark_bar().encode(
    x='a:O',
    y='b:Q',
    color='a',
).properties(title='Our Data')

# save chart information using .json extension-
chart.save('data.json')

# another option is to use .show() method-
# chart.show()
```



In example program, we have seen that we used ‘`chart.save(data.json)`’ and ‘`chart.show()`’.

chart.save(‘filename.json’)-

- It saves the data content in a file with ‘`.json`’ extension, this content we have to copy and paste in the left side of the editor (vega-editor opened in the link). We can see the chart (plots) in the right side.

chart.show()-

- It shows the chart in a default browser directly when you run the program.

➤ *You can use any of the above methods*