

Deneb Cheat Sheet

ENTERPRISE DNA

Deneb website: https://deneb-viz.qithub.io/ Vega-Lite website: https://vega.qithub.io/vega-lite/

Enterprise DNA Forum Deneb Showcase category: https://forum.enterprisedna.co/c/deneb-showcase/58

GENERAL

The required 3 basic **block** types of a Deneb/Vega-Lite specification are data, mark, and encoding:

```
"data": {"name": "dataset"},

"mark": {"type": "bar"},

"encoding": {
    "y": {"field": "Country", "type": "nominal"},
    "x": {"field": "Total Sales", "type": "quantitative"},
    "color": {"field": "Channel", "type": "nominal"} }
```

There are several mark types in Vega-Lite, including:

- common: bar, line, arc, circle, text, area
- other: point, rect, rule, square, tick, geoshape
- composite: boxplot, errorband, errorbar

https://bit.ly/3Bm9xuG

There are 4 field-mapping types:

- nominal: the category of the data (use if string)
- quantitative: the value of the data (use if numeric)
- temporal: the value of the data (use if date)
- ordinal: a ranked order for data sorting

https://bit.ly/3Bvx7VM

SORTING

To sort an axis by a field, add a **sort** block to its' encoding block: "sort": {

```
"op": "sum",
"field": "Total Sales",
"order": "descending"}
```

FORMATTING

To **format** a data value using Power BI format strings, add to its' definition in the encoding block:

```
"encoding": {
    "x": {
        "field": "Total Sales",
        "type": "quantitative",
        "axis": {
            "format": "#0,,.0M",
            "formatType": "pbiFormat" } } }
```

https://bit.ly/3cUwwD3

TOOLTIPS

To enable **tooltips**, add a property to the mark block:

```
"mark": {"type": "bar", "tooltip": true}
```

To customize tooltips, add a **tooltip** block to the encoding block:

https://bit.ly/3bppOnT

LAYERS

To create a visual using multiple overlapping marks, create a **layer** consisting of an array of specifications, one for each mark:

```
"data": {"name": "dataset"},
"layer": [
   "name": "TOTAL SALES",
   "mark": {
    "type": "bar",
     "size": {"expr": "bandwidth('y')"} },
   "encoding": {
     "x": {"field": "Total Sales", "type": "quantitative"},
     "y": {"field": "Country", "type": "ordinal"},
     "color": {"value": "blue" } }
},
   "name": "EXPORT SALES",
   "mark": {
     "type": "bar",
    "size": {"expr": "bandwidth('y') * 0.5" } },
   "encoding": {
     "x": {"field": "Export Sales", "type": "quantitative"},
     "y": {"field": "Country", "type": "ordinal"},
     "color": {"value": "red" } }
```

To **share** encoding between multiple marks, move the relevant sections of the encoding block outside the layer block:

```
"data": {"name": "dataset"},
"encoding": {
 "y": {"field": "Country", "type": "ordinal"} },
"layer": [
   "name": "TOTAL SALES",
   "mark": {
     "type": "bar".
     "size": {"expr": "bandwidth('y')"} },
     "x": {"field": "Total Sales", "type": "quantitative"},
     "color": {"value": "blue" } }
},
   "name": "EXPORT SALES",
  "mark": {
     "type": "bar",
    "size": {"expr": "bandwidth('y') * 0.5" } },
   "encoding": {
     "x": {"field": "Export Sales", "type": "quantitative"},
     "color": {"value": "red" } }
```

(NOTE 1: Layer order is respected by Deneb/Vega-Lite, and layers are rendered in order from first [i.e., bottom] to last [i.e., top])

(NOTE 2: Encoding in Deneb/Vega-Lite follows a cascading model where the inner encoding **inherits** the outer encoding, but can also **override** the outer encoding)

OPACITY

To set the transparency of a visual:

```
• adjust the opacity of the mark
"mark": {"type": "bar", "opacity": 0.3}
```

OF

• add an opacity block to the encoding block:

```
"encoding": {
    "x": {"field": "Total Sales_highlight" },
    "opacity": {
        "condition": {
            "test": {"field": "_selected_", "equal": "off"},
            "value": 0},
        "value": 1} } }
```

https://bit.ly/3DxRD9c

TITLE

To set a visual title, add a **title** block:

```
"title": {
  "text": "Total Sales by Country",
  "anchor": "start",
  "align": "left" }
```

WIDGET

To create an **input widget**, add to the parameters block:

```
{ "name": "tension", "value": 0,
"bind" ("input": "range", "min": 0, "max": 1, "step": 0.1} }
https://bit.ly/3zlP4is
```

TEMPLATE

To export a **template** of your Deneb visual for reuse:

- click the Generate JSON Template icon in the Visual Editor toolbar
- complete the fields on the Template Information and Dataset (Columns and Measures) tabs
- copy all JSON code from the Generated Template tab and paste into a text editor, saving the file with a .json extension

To use an existing **template** in your Deneb visual:

- when first editing a Deneb visual, click the Import From Template tab in the Create New Specification dialog
- click the Select JSON Template link in the Create New Specification dialog
- browse to the saved location of the desired .json file and click on the .json file

https://bit.ly/3Seugge

EDITOR FONT SIZE

To set the font size in the Deneb Visual Editor, use the JSON Editor Font Size drop-down in the Properties Pane

V1.0 | October 2022

Prepared by: Greq Philps



Deneb Cheat Sheet

ENTERPRISE DNA

Deneb website: https://deneb-viz.github.io/ Vega-Lite website: https://vega.github.io/vega-lite/

Enterprise DNA Forum Deneb Showcase category: https://forum.enterprisedna.co/c/deneb-showcase/58

LINKED CHARTS

To link 2 charts together, add a selection brush to the first visual: "selection": {"brush": {
"type": "interval", "encodings": ["x"] } }

And use the selection in a filter transform in the second visual: "transform": [{ "filter": { "param": "brush" } }] https://bit.ly/3JhxaGC

EXTEND DATA/ENHANCE VISUAL

To extend a dataset with derived fields, add a transform block:

To refer to an existing dataset field, use the format datum['field name'] (datum.field_name can be used only if there are no spaces in the field name)

```
"transform": [
{"calculate": "2*3.14159*datum['Ring1 Percent']",
    "as": "Ring1 Radians" },
{"calculate": "2*3.14159*datum['Ring2 Percent']",
    "as": "Ring2 Radians" },
{"calculate": "2*3.14159*datum['Ring3 Percent']",
    "as": "Ring3 Radians" } ]
```

https://bit.ly/3ziGPrU

(Many additional transformations are available, including aggregate, filter, flatten, fold, etc.)

https://bit.ly/3Bu4Q21

To enhance a visual with named values or expressions, add a params block:

Add common math **constants** to expressions:

NAMED COLOURS

To use a **named** colour:

```
"mark": {
  "type": "area",
  "line": { "color": "darkgreen" } }
```

To use a built-in Vega-Lite colour **scheme**:

```
"color": {
   "field": "series",
   "scale": { "scheme": "category20b" } }
```

(There are several schemes available, including category20b, dark2, set1, set2, etc.)

https://bit.ly/3bpoK3l

THEME COLOURS

To use a specific Power BI **theme** colour, adjust the theme index (zero-based, so theme colour - 1):

```
"color": { "value": { "expr": "pbiColor(0)" } }
https://bit.lv/3vp9llg
```

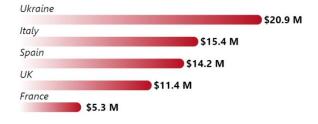
GRADIENT COLOURS

To add Power BI gradient colours, add to the color block: "scale": { "scheme": "pbiColorLinear" } https://bit.ly/3OJXNoE

There are 4 integrations with the Power BI theme colours in Deneb:

- pbiColorNominal: matches the current theme colours
- pbiColorOrdinal: uses a ramped scale from the MAX to MIN divergent colour of the current theme, and EXCLUDES the MIDDLE colour
- pbiColorLinear: uses an interpolated gradient from the MAX to MIN divergent colour of the current theme, and EXCLUDES the MIDDLE colour
- pbiColorDivergent: same as pbiColorLinear, but INCLUDES the MIDDLE colour

https://bit.ly/3bdEkPA



LINE CHARTS

To set the smoothing and tension of a line chart, add to the line mark block:

```
"interpolate": "cardinal", "tension": 0.7
```

 options include basis, monotone, natural, step, etc. https://bit.ly/3RRZSRL

To set the label format for a temporal axis, "axis": { "format": "%y-%b-%d" } https://bit.ly/3cVA480

BAR CHARTS

To turn a bar chart into a **column** chart, exchange the X and Y encoding:

```
"encoding": {
   "x": { "field": "Country", "type": "nominal" },
   "y": { "field": "Total Sales", "type": "quantitative" } }
```

To set the column labels to **horizontal**, add to X encoding block:

```
axis": {
  "labelAngle": 0 }
```

To **round** bar corners, add to bar mark:

```
"cornerRadius": 10 (all corners) OR
"cornerRadiusEnd": 25 (ends only)
https://bit.ly/3DzuXpi
```

To turn a bar chart into a **stacked** bar chart, add a color block to the encoding block:

```
"color": {
  "field": "Channel",
  "type": "nominal" }
https://bit.ly/3zKU4D3
```

To turn a stacked bar chart into a 100% stacked bar chart, add to the Y encoding block:

```
"stack": "normalize" https://bit.ly/3oJKx9e
```

CIRCULAR CHARTS

To set the size (outer radius) of a **pie** chart, add a radius or outer radius property to the arc mark block:

```
"mark": {
  "type": "arc",
  "outerRadius": 100 }
```

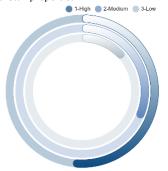
To turn a pie chart into a **donut** chart, add a radius2 or inner radius property to the arc mark block:

```
"innerRadius": 75
https://bit.ly/3cUYRcj
```

To set the start and end of an **arc** segment (in radians; 0 = "north/up"), use the theta and theta2 properties.

https://bit.lv/3vvE9X0

```
"mark": {
    "type": "arc",
    "radius": 200,
    "radius2": 100,
    "theta": 0,
    "theta2": 5.9 }
```



V1.0 | October 2022



Deneb Cheat Sheet

ENTERPRISE DNA

Deneb website: https://deneb-viz.github.io/
Vega-Lite website: https://vega.github.io/vega-lite/

Enterprise DNA Forum Deneb Showcase category: https://forum.enterprisedna.co/c/deneb-showcase/58

VIEW COMPOSITION

There are 4 different types of composite views:

 layering: a set of overlapping specifications (described in more detail on the GENERAL page)

```
{ << SPECIFICATION 1 >> },
{ << SPECIFICATION 2 >> } ]
```

 concatenation: a set of specifications concatenated vertically (vconcat), horizontally (hconcat), or so they wrap (concat)

```
{ << SPECIFICATION 1 >> },
{ << SPECIFICATION 2 >> } ]
```

 faceting: a set of repeating marks each with a subset of the data (like small multiples) using a common specification

```
"encoding": {
   "facet": {
      "field": "Country", "type": "ordinal", "columns": 2 } )
   NOTE: Faceting doesn't work in an encoding channel for
   layers; is often an issue when converting a single view to a
   layered view with facets in them
```

 repeating: a set of repeating marks each with the full dataset using a common specification

```
"data": {"name": "dataset"},
   "<mark>repeat</mark>": [
     "Horsepower", "Acceleration", "Displacement"],
   "columns": 2,
   "<mark>spec</mark>":
     "mark": "bar",
     "encoding": {
        "x": {
          "field": {"repeat": "repeat"}, "bin": true},
        "y": {
          "aggregate": "count"},
          "field": "Origin",
          "scale": {"scheme": "pbiColorNominal"} }
                                                                     Europe
Count of Records
     40 60 80 100120140160180200220240
                                          10 15 20 25 30 35 40 45 50
                                           Miles_per_Gallon (binned)
          Horsepower (binned)
                                     50
   50
```

Displacement (binned)

12 14 16 18 20 22 24 26

Acceleration (binned)

CONDITIONS

To use a condition to set the value of a property:

- To refer to an existing dataset field, use the format datum['field name'] (datum.field_name can be used only if there are no spaces in the field name)
- Colour of a mark (based on tests):

Colour of a mark (based on parameter):

```
"color": { "condition": {
    "param": "brush", "field": "Cylinders", "type":
"ordinal" },
    "value": "grey" }
```

• Size of a mark (based on mouse selection):

```
"size": {"condition": {
    "param": "paintbrush", "value": 600 }, "value": 50 }
```

IF-THEN-ELSE

To add an if-then-else calculation, use the (ifTest ? thenValue : elseValue) form (ternary operator):

```
"mark": {
    "type": "bar",
    "color": {"expr": "datum['Sales'] > 10 ? 'green' : 'red'"
}},
...
```

EXPRESSIONS

To use a parameter, add an expr block:

```
"params": [{"name": "my_colour", "value": "blue"}],
"mark": {"type": "bar", "color": {"expr": "my_colour" }},
...
```

DENEB/VEGA-LITE OBJECT MODEL

As Deneb/Vega-Lite is so flexible, a comprehensive **object model** (DVLOM?) would be quite large with many optional and alternate paths, and it's complexity might make it less useful. A partial, simplified, and high-level object model showing only a few of the common objects in Deneb/Vega-Lite can provide an overview.

A few notes:

- Many key-value pairs can be used in multiple locations (e.g., opacity in mark or encoding, etc.)
- Many key-value pairs are commonly used with only specific marks (e.g., the "text" encoding is used with a "text" mark, etc.)
- Aside from the common data, mark, and encoding blocks, a specification does not require many of the key-value pairs (thus they are optional) (e.g., formatType, format, title, etc.)

