D3 examples from Adam Janes his gist repo

<https://gist.github.com/adamjanes>

D3 v5 code examples – cannot see how to get this in the correct file structure…

<https://observablehq.com/@d3/gallery>

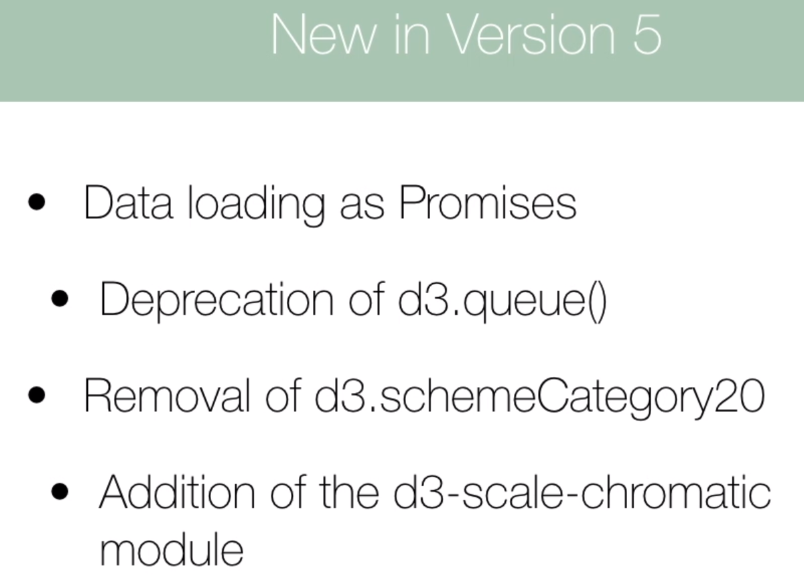
This one has better format use it for now

<https://blockbuilder.org/search?d3version=v5>

Another one to check

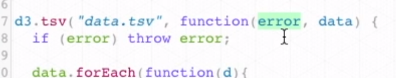
<https://vizhub.com/>

D3 JS changes with version 5



1. Change the code over to v5, if it breaks look at the data loading function first. This is handled differently in v5. Use promises in v5

V4



   d3.csv("data/tree.csv", *function* (*error*, *data*) {

      if (error) throw error;

    });

V5



    d3.csv("data/tree.csv")

      .then((*data*) *=>* {

        console.log(data);

      })

      .catch(*function* (*error*) {

        console.log(error);

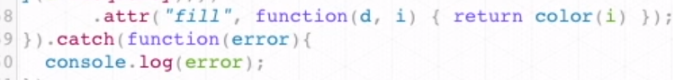
      });

1. Replace schemeCategory20 with some other colors scheme

<https://hub.packtpub.com/d3-5-0-is-out/>

1. 

Add error handling for the data load function at the end



Example syntax for v4

    v4 syntax

    d3.tsv(

      "data/area.tsv",

      // like a for each for a tsv file

*function* (*d*) {

        d.date = parseTime(d.date);

        d.close = +d.close;

        return d;

      },

*function* (*error*, *data*) {

        if (error) throw error;

        x.domain(

          d3.extent(data, *function* (*d*) {

            return d.date;

          })

        );

        y.domain([

          0,

          d3.max(data, *function* (*d*) {

            return d.close;

          }),

        ]);

        g.append("path").attr("fill", "steelblue").attr("d", area(data));

        g.append("g")

          .attr("transform", "translate(0," + height + ")")

          .call(d3.axisBottom(x));

        g.append("g")

          .call(d3.axisLeft(y))

          .append("text")

          .attr("fill", "#000")

          .attr("transform", "rotate(-90)")

          .attr("y", 6)

          .attr("dy", "0.71em")

          .attr("text-anchor", "end")

          .text("Price ($)");

      }

    );

Made to version 5

    d3.tsv("data/area.tsv").then((*data*) *=>* {

      // console.log(data);

      data.forEach((*d*) *=>* {

        d.date = parseTime(d.date);

        d.close = +d.close;

        return d;

      });

      x.domain(

        d3.extent(data, *function* (*d*) {

          return d.date;

        })

      );

      y.domain([

        0,

        d3.max(data, *function* (*d*) {

          return d.close;

        }),

      ]);

      g.append("path").attr("fill", "steelblue").attr("d", area(data));

      g.append("g")

        .attr("transform", "translate(0," + height + ")")

        .call(d3.axisBottom(x));

      g.append("g")

        .call(d3.axisLeft(y))

        .append("text")

        .attr("fill", "#000")

        .attr("transform", "rotate(-90)")

        .attr("y", 6)

        .attr("dy", "0.71em")

        .attr("text-anchor", "end");

    });

This is a code camp tutorial not part of the class but looks good.

<https://www.freecodecamp.org/news/how-to-work-with-d3-jss-general-update-pattern-8adce8d55418/>

Adams answer from lecture 64 –

[Adam](https://www.udemy.com/user/adam-janes/) — Instructor  
2 days ago

**0**

Hey Ar,

Check out this example from the [D3 documentation](https://github.com/d3/d3-fetch/tree/v1.1.2):

1. **d3.dsv(",", "test.csv", function(d) {**
2. **return {**
3. **year: new Date(+d.Year, 0, 1), // convert "Year" column to Date**
4. **make: d.Make,**
5. **model: d.Model,**
6. **length: +d.Length // convert "Length" column to number**
7. **};**
8. **}).then(function(data) {**
9. **console.log(data);**
10. **});**

**The format is pretty much the same from looking at the d3.json() signature:**

**d3.json(*input*[, *init*])**

**I think that this should work similarly to how v4 did it. You can pass an optional function as the second argument to d3.json(), which can apply some transformation to each row of data.**

**Alternatively, you can still use what we tended to do in this coure:**

1. **d3.json('path/to/file.json').then(function(data) {**
2. **data.forEach(function(d){**
3. **// make change here**
4. **})**
5. **})**