

# D3 v5 changes

- "only a few non-backwards-compatible changes"
- no more `d3.schemeCategory20`, uses `d3-scale-chromatic` instead
- loading data uses *promises* instead of *callbacks*
- a few other new things
- <https://github.com/d3/d3/blob/master/CHANGES.md>

# Goodbye, d3.schemeCategory20\*



<https://bl.ocks.org/pstuffa/3393ff2711a53975040077b7453781a9>

# Hello, d3-scale-chromatic

## API Reference

---

### Categorical

# d3.schemeCategory10 <>



An array of ten categorical colors represented as RGB hexadecimal strings.

# d3.schemeAccent <>



An array of eight categorical colors represented as RGB hexadecimal strings.

# d3.schemeDark2 <>



An array of eight categorical colors represented as RGB hexadecimal strings.

<https://github.com/d3/d3-scale-chromatic>

# Asynchronous Programming

## Events

`.on("click", function () {...})`

**Callbacks** `d3.csv()` v4

**Promises** `d3.csv()` v5

# Callbacks

```
d3.csv("mydata.csv", function () {  
    all the stuff that uses the data  
});
```

## Multiple line graphs with labels

[bl.ocks.org/d3noob/8603837](http://bl.ocks.org/d3noob/8603837)

## Multi-line graph 2 with v4: Colours

<https://bl.ocks.org/d3noob/ae9786c26d6a821eeefeabe60dec350a9>

# Loading files

- 1. Use a local server**
- 2. Work on [blockbuilder.org](https://blockbuilder.org)**

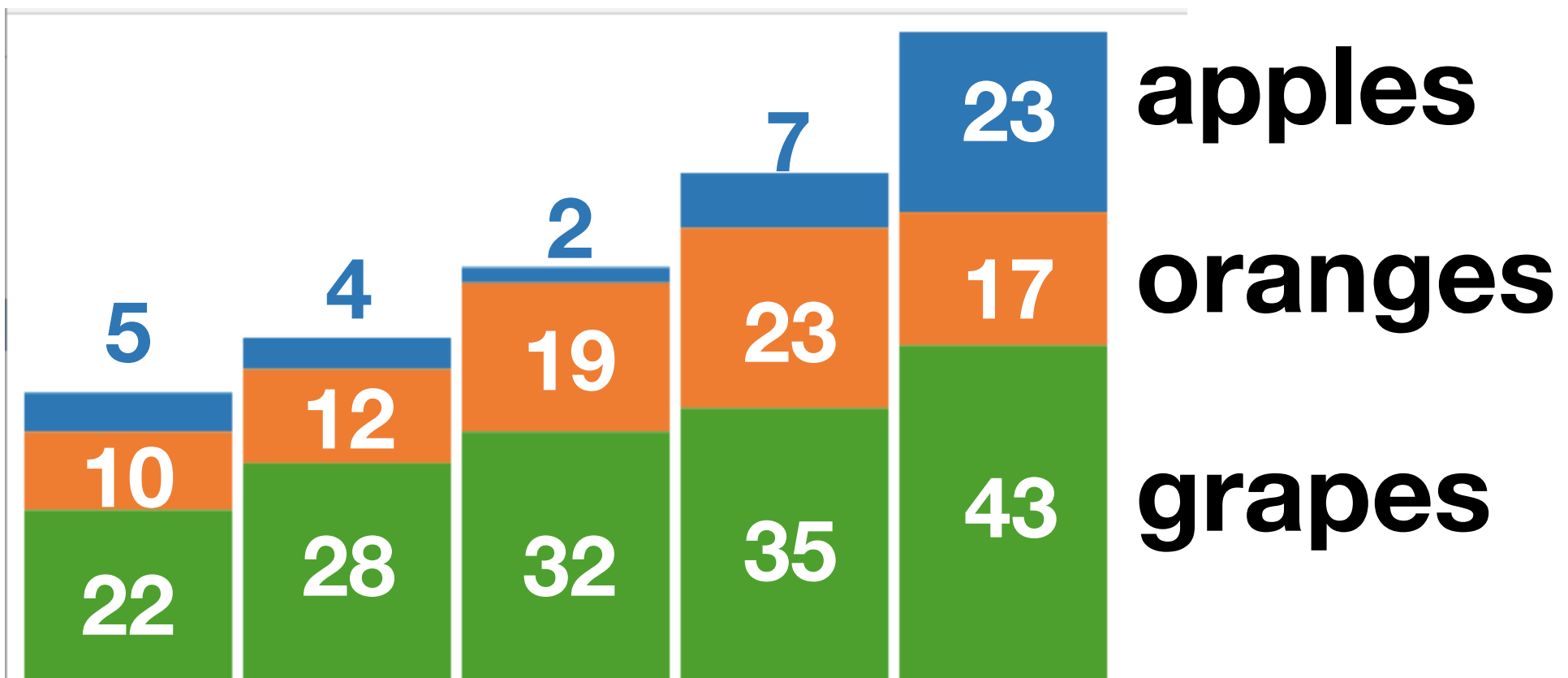
# Layouts (Ch. 13)

**reorganize your data into a more convenient form for the type of graph you want to draw**

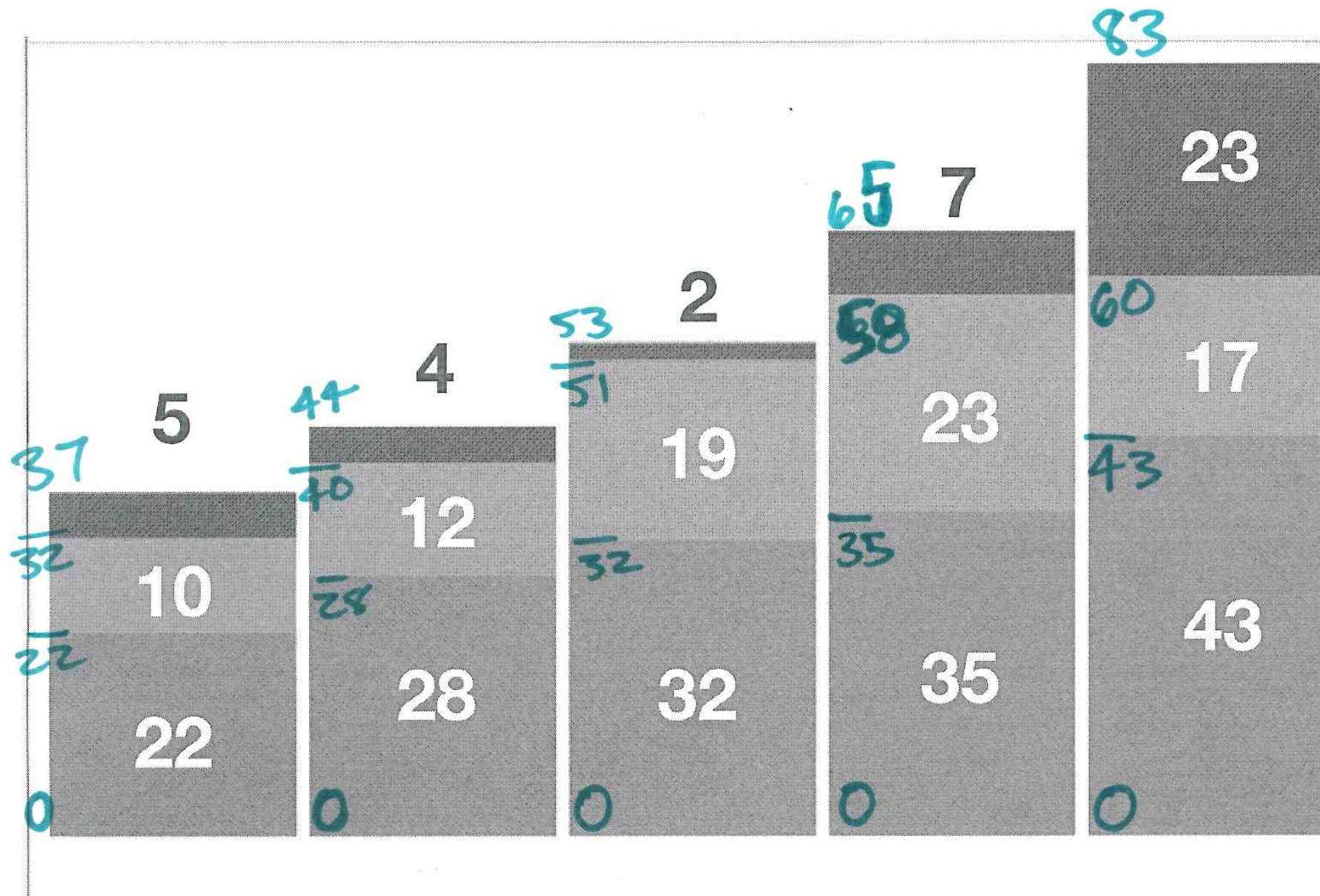
**d3.stack()**

## d3.stack()

```
var dataset = [  
  { apples: 5, oranges: 10, grapes: 22 },  
  { apples: 4, oranges: 12, grapes: 28 },  
  { apples: 2, oranges: 19, grapes: 32 },  
  { apples: 7, oranges: 23, grapes: 35 },  
  { apples: 23, oranges: 17, grapes: 43 }  
];
```







**apples**  
Series[0]

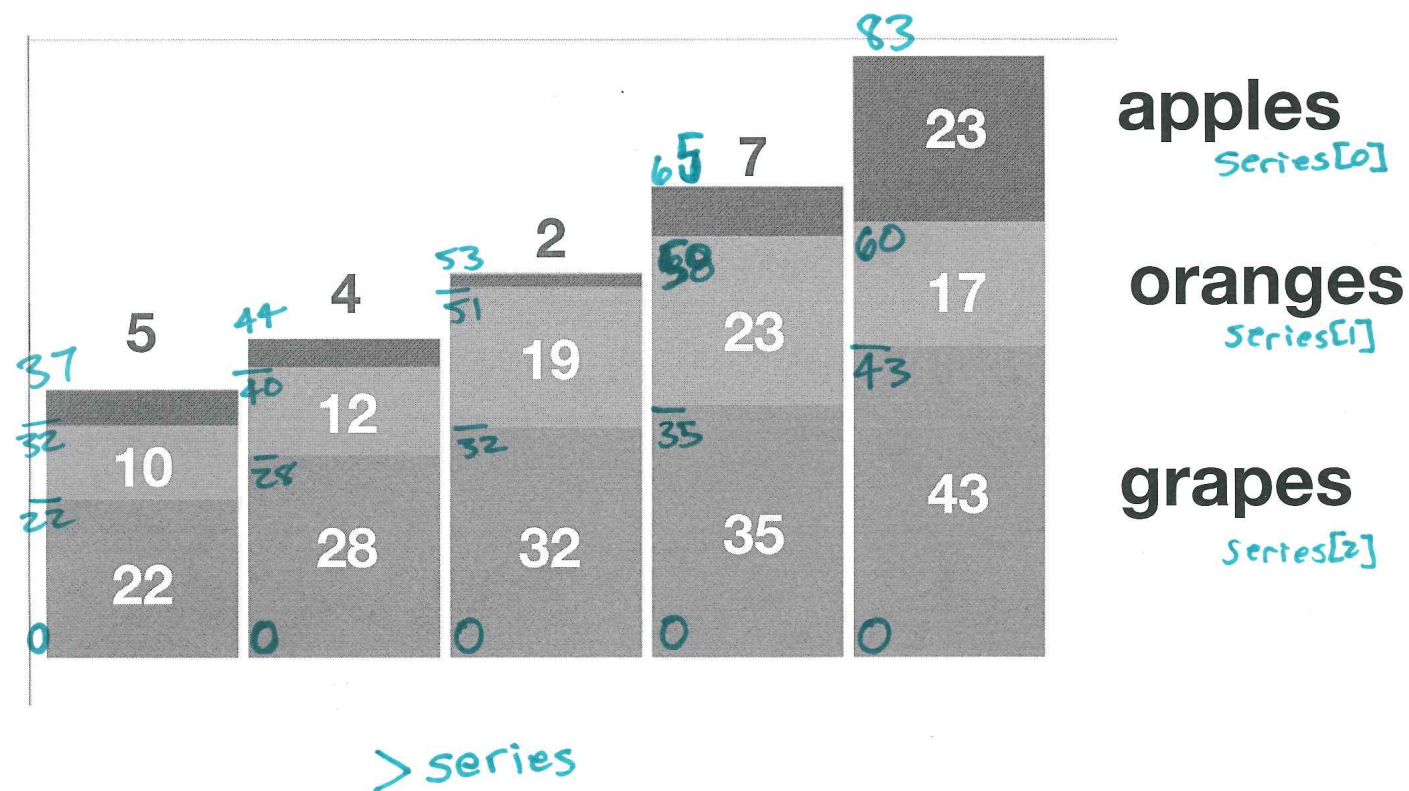
**oranges**  
Series[1]

**grapes**  
Series[2]

> series

## d3.stack()

```
var dataset = [  
  { apples: 5, oranges: 10, grapes: 22 },  
  { apples: 4, oranges: 12, grapes: 28 },  
  { apples: 2, oranges: 19, grapes: 32 },  
  { apples: 7, oranges: 23, grapes: 35 },  
  { apples: 23, oranges: 17, grapes: 43 }  
];  
  
var stack = d3.stack()  
  .keys( [ "apples", "oranges", "grapes" ] )  
  .order(d3.stackOrderDescending);  
  
var series = stack(dataset);
```



> series

< ▼ (3) [Array(5), Array(5), Array(5)]

▼ 0: Array(5)

- ▶ 0: (2) [32, 37, data: {...}]
- ▶ 1: (2) [40, 44, data: {...}]
- ▶ 2: (2) [51, 53, data: {...}]
- ▶ 3: (2) [58, 65, data: {...}]
- ▶ 4: (2) [60, 83, data: {...}]

index: 2

key: "apples"

length: 5

▶ \_\_proto\_\_: Array(0)

▼ 1: Array(5)

- ▶ 0: (2) [22, 32, data: {...}]
- ▶ 1: (2) [28, 40, data: {...}]
- ▶ 2: (2) [32, 51, data: {...}]
- ▶ 3: (2) [35, 58, data: {...}]
- ▶ 4: (2) [43, 60, data: {...}]

index: 1

key: "oranges"

length: 5

▶ \_\_proto\_\_: Array(0)

▼ 2: Array(5)

- ▶ 0: (2) [0, 22, data: {...}]
- ▶ 1: (2) [0, 28, data: {...}]
- ▶ 2: (2) [0, 32, data: {...}]
- ▶ 3: (2) [0, 35, data: {...}]
- ▶ 4: (2) [0, 43, data: {...}]

index: 0

key: "grapes"

length: 5

▶ \_\_proto\_\_: Array(0)



# Now what?

## Step 1:

**Append a group for each array  
(that is, each fruit)**

```
var groups = svg.selectAll("g")
    .data(series)
    .enter()
    .append("g")
    .style("fill", function(d,i) {
        return colors(i);
    });
```

# One fruit per group

<g>

<g>

<g>

```
> series
< ▼ (3) [Array(5), Array(5), Array(5)]
  ▼ 0: Array(5)
    ► 0: (2) [32, 37, data: {...}]
    ► 1: (2) [40, 44, data: {...}]
    ► 2: (2) [51, 53, data: {...}]
    ► 3: (2) [58, 65, data: {...}]
    ► 4: (2) [60, 83, data: {...}]
    index: 2
    key: "apples"
    length: 5
    ► __proto__: Array(0)
  ▼ 1: Array(5)
    ► 0: (2) [22, 32, data: {...}]
    ► 1: (2) [28, 40, data: {...}]
    ► 2: (2) [32, 51, data: {...}]
    ► 3: (2) [35, 58, data: {...}]
    ► 4: (2) [43, 60, data: {...}]
    index: 1
    key: "oranges"
    length: 5
    ► __proto__: Array(0)
  ▼ 2: Array(5)
    ► 0: (2) [0, 22, data: {...}]
    ► 1: (2) [0, 28, data: {...}]
    ► 2: (2) [0, 32, data: {...}]
    ► 3: (2) [0, 35, data: {...}]
    ► 4: (2) [0, 43, data: {...}]
    index: 0
    key: "grapes"
    length: 5
    ► __proto__: Array(0)
```

> d3.select("g").data()

```
< ▼ [Array(5)] 1
  ▼ 0: Array(5)
    ► 0: (2) [32, 37, data: {...}]
    ► 1: (2) [40, 44, data: {...}]
    ► 2: (2) [51, 53, data: {...}]
    ► 3: (2) [58, 65, data: {...}]
    ► 4: (2) [60, 83, data: {...}]
```

## Step 2: Draw the rects

```
var rects = groups.selectAll("rect")
  .data(d => d) <-- Bind the group data
  .enter()
  .append("rect")
  .attr("y", d => yScale(d[0]))
  .attr("height", d =>
    yScale(d[1]) - yScale(d[0]))
  .attr(
    ...
  );
});
```