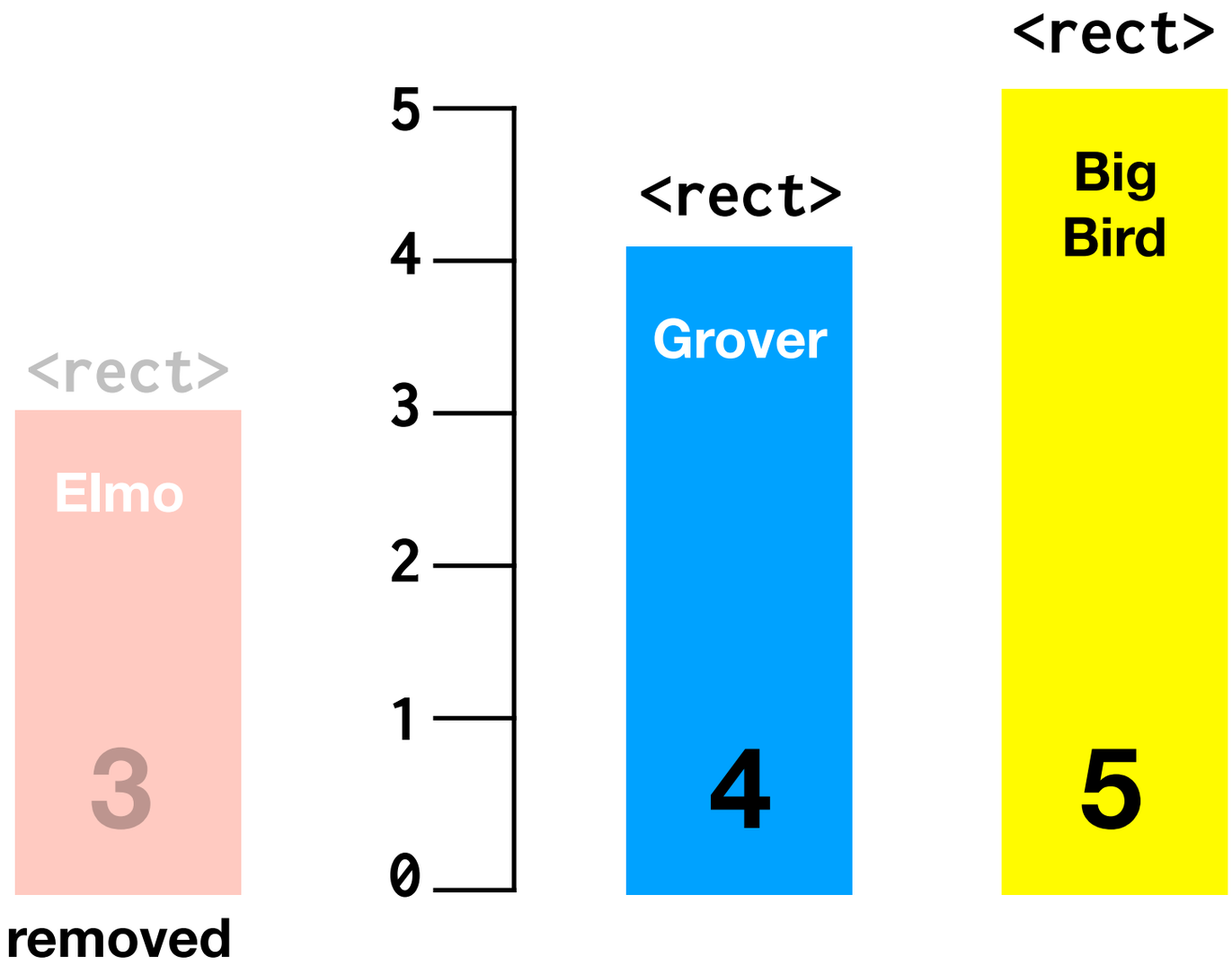
Object constancy

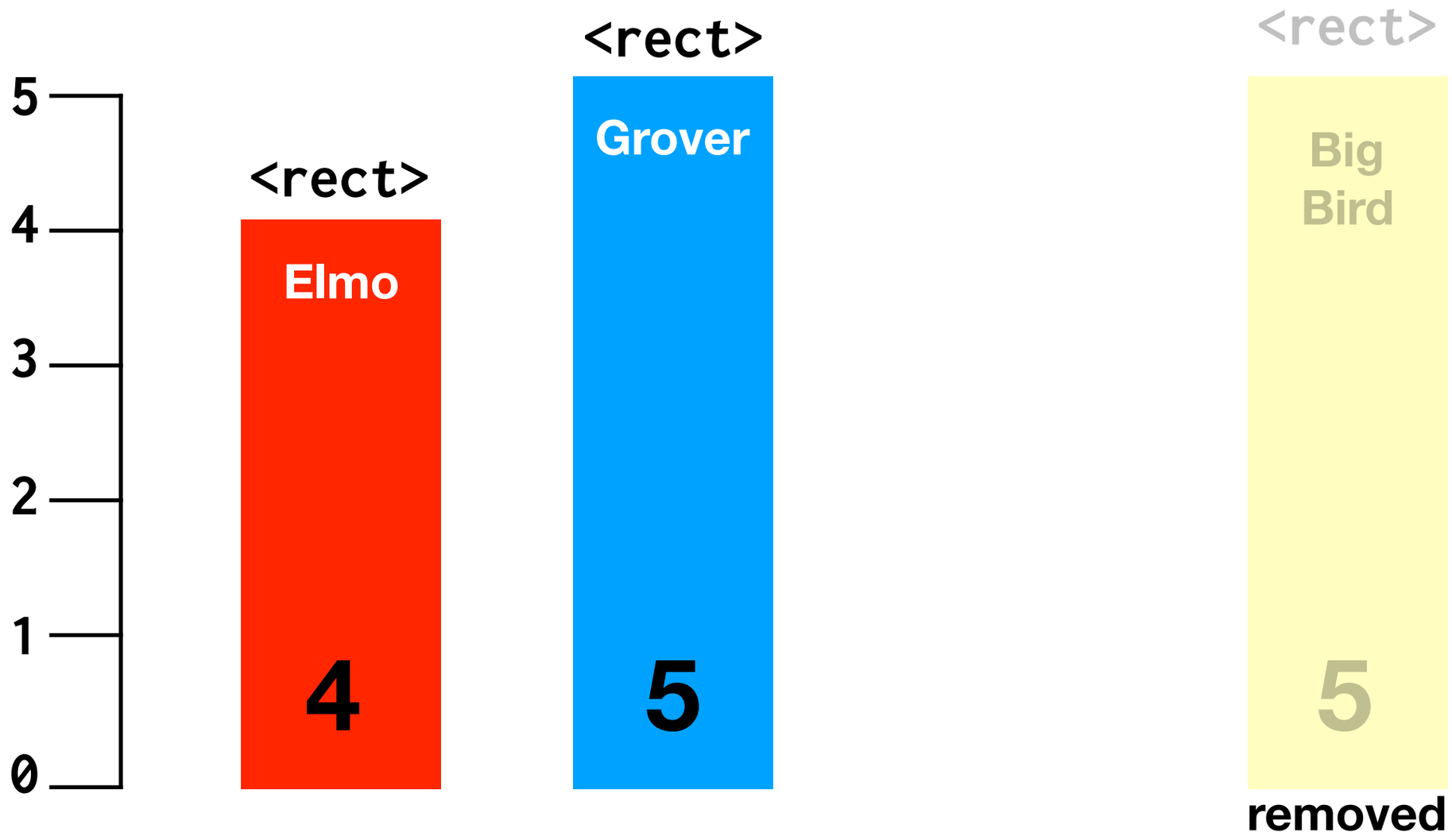
Starting positions

Object constancy maintained



Bars shift to the left one

Object constancy not maintained



Bars shift to the left one

Object constancy

- **Join data by key, not in order of SVG elements**

ex.

```
var bardata = [{key: 0, value: 300},
               {key: 1, value: 100},
               {key: 2, value: 150},
               {key: 3, value: 220},
               {key: 4, value: 70},
               {key: 5, value: 270}];
```

Creating array of keys from dataset

```
> var bardata = [23, 34, 123, 29]
    .map((d, i) => ({key: i, value: d}));

> bardata
```

```
(4) [{…}, {…}, {…}, {…}]
  0: {key: 0, value: 23}
  1: {key: 1, value: 34}
  2: {key: 2, value: 123}
  3: {key: 3, value: 29}
```

Object constancy

- **Specify the key when joining data:**

 # selection.**data**([*data*[, *key*]]) <>

```
.data(bardata, d => d.key);
```

- **Replace every "`d`" with "`d.value`"**

- **Use a key when adding data:**

```
bardata.push({ key: newkey,
               value: newvalue });
```

Do not use `d3.keys()`

Object constancy

key: 0 value: 23	key: 1 value: 34	key: 2 value: 123	key: 3 value: 29

```
> d3.select("svg")
   .selectAll("rect")
   .data([{key: 0, value: 200},
          {key: 3, value: 100}],
          d => d.key)
   .exit()
   .remove();
```

What will happen?

Object constancy

key: 0
value: 23

key: 1
value: 34

key: 2
value: 123

key: 3
value: 29

```
> d3.select("svg")
   .selectAll("rect")
   .data([{key: 0, value: 200},
          {key: 3, value: 100}],
          d => d.key)
   .exit()
   .remove();
```

key: 0
value: 200

key: 3
value: 100

Object constancy

key: 0 value: 23	key: 1 value: 34	key: 2 value: 123	key: 3 value: 29

```
> d3.select("svg")
   .selectAll("rect")
   .data([{key: 10, value: 234},
          {key: 13, value: 57}],
          d => d.key)
   .exit()
   .remove();
```

What will happen?

Object constancy

key: 0 value: 23	key: 1 value: 34	key: 2 value: 123	key: 3 value: 29

```
> d3.select("svg")
   .selectAll("rect")
   .data([{key: 10, value: 234},
          {key: 13, value: 57}],
          d => d.key)
   .exit()
   .remove();
```

No matches: all elements disappear.

Enter selection contains two elements.

Set up a local server

- **SimpleHTTPServer (pp. 64-65)**
- **http-server**

https://www.npmjs.com/package/http-server

first install npm: https://www.npmjs.com/