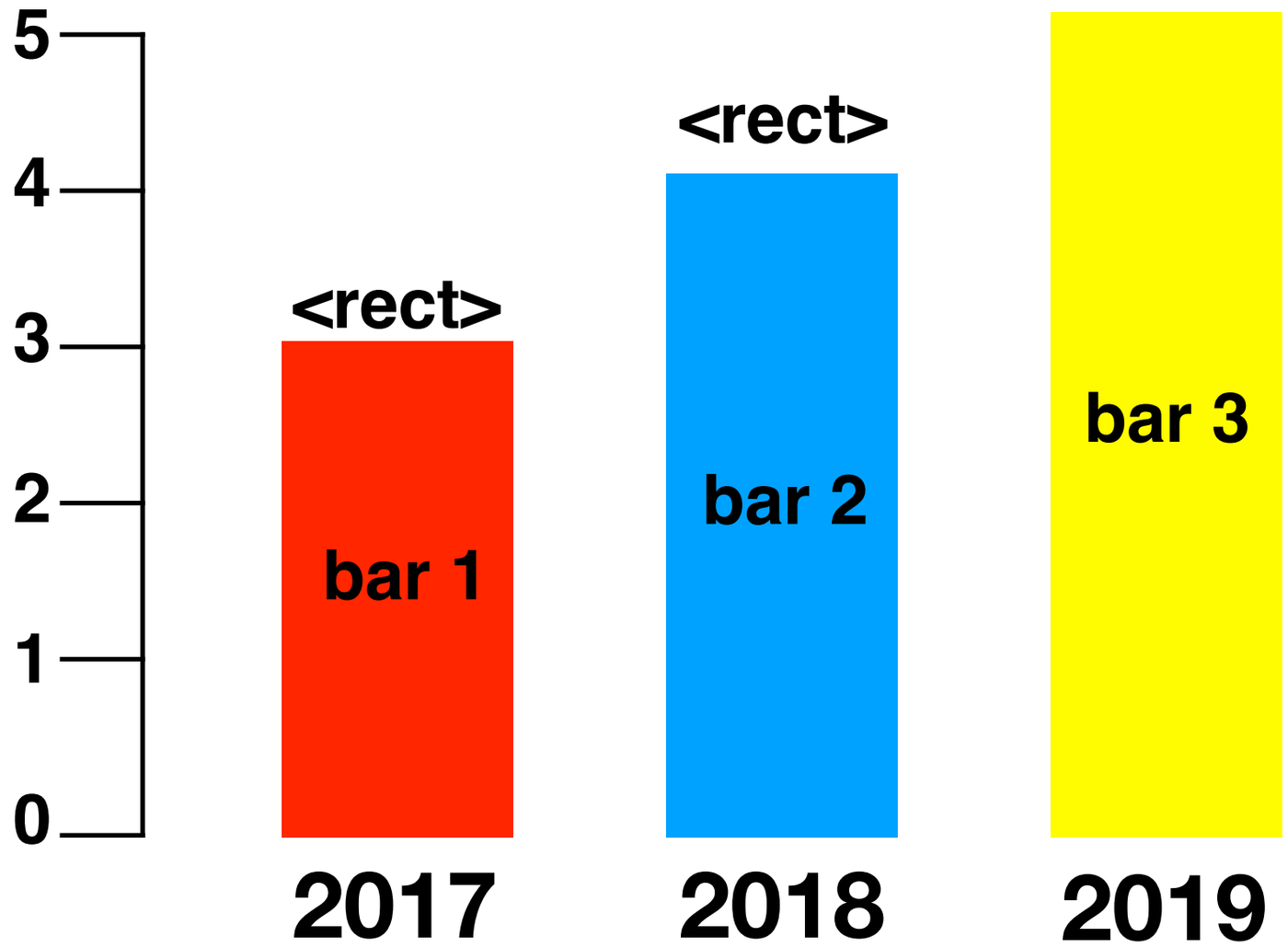
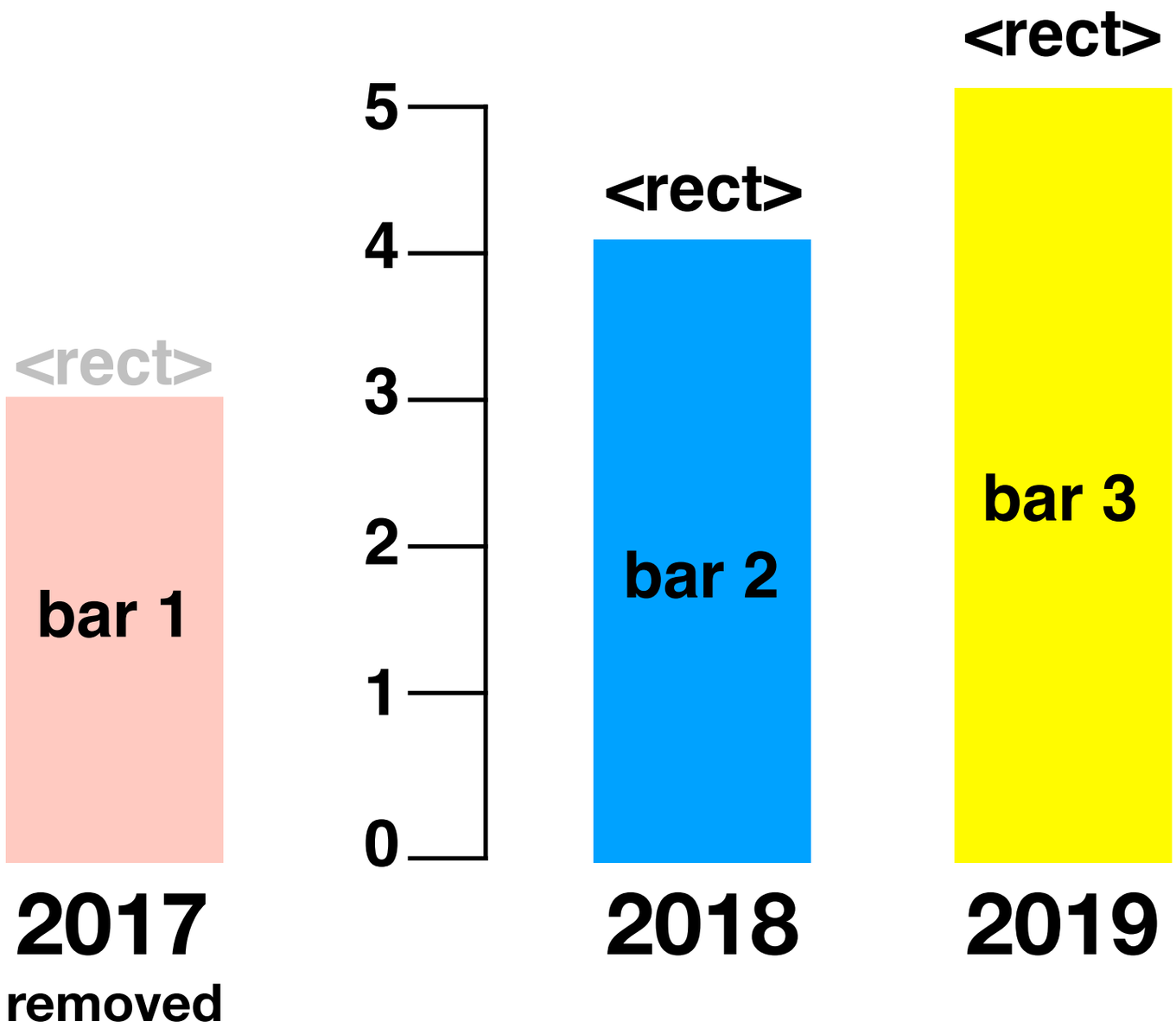
Object Constancy

Initial setup

Goal: remove 2017

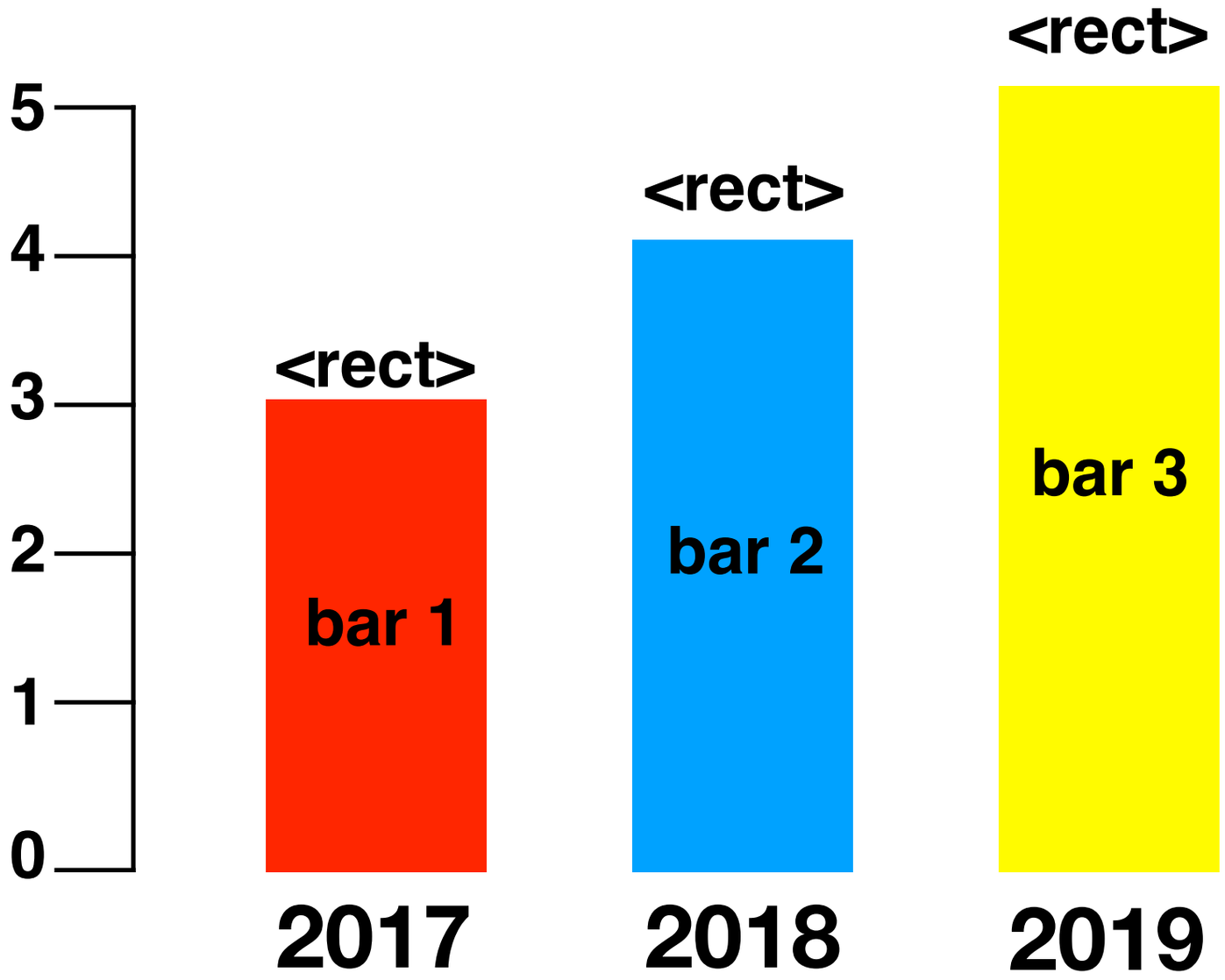


Object constancy maintained

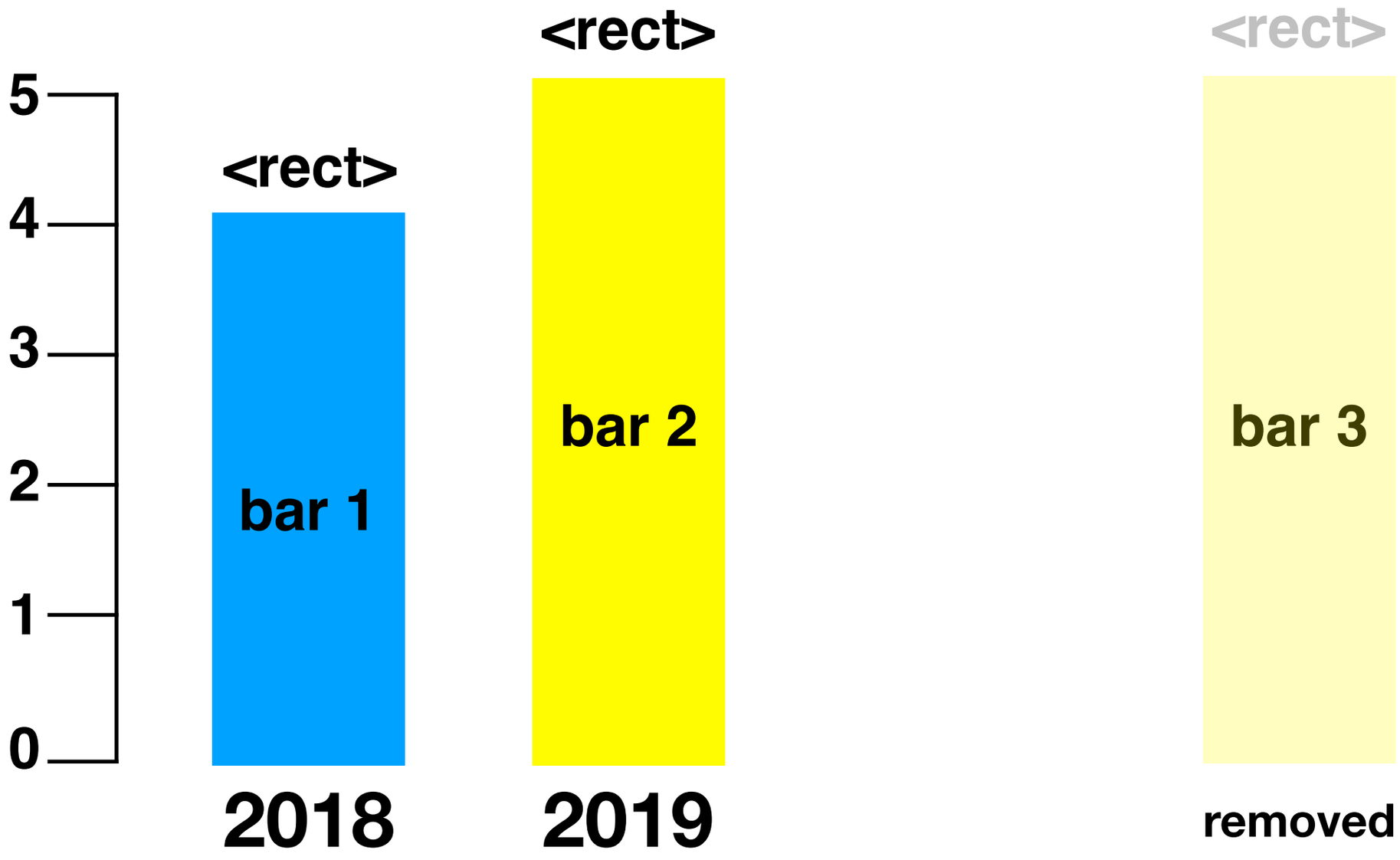


- red (2017) is removed on left side
- other bars (2018, 2019) shift left

Initial setup

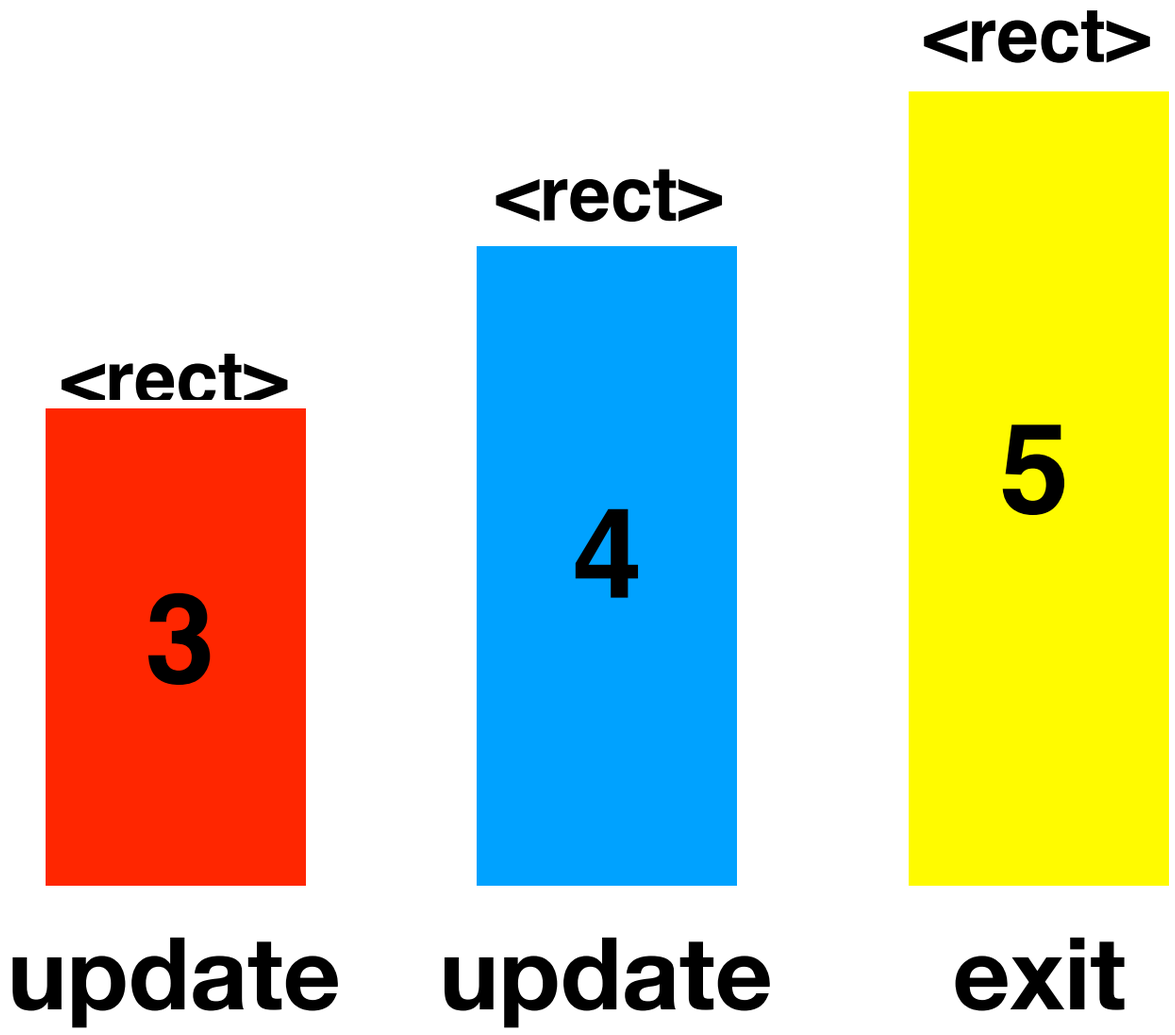


Object constancy not maintained



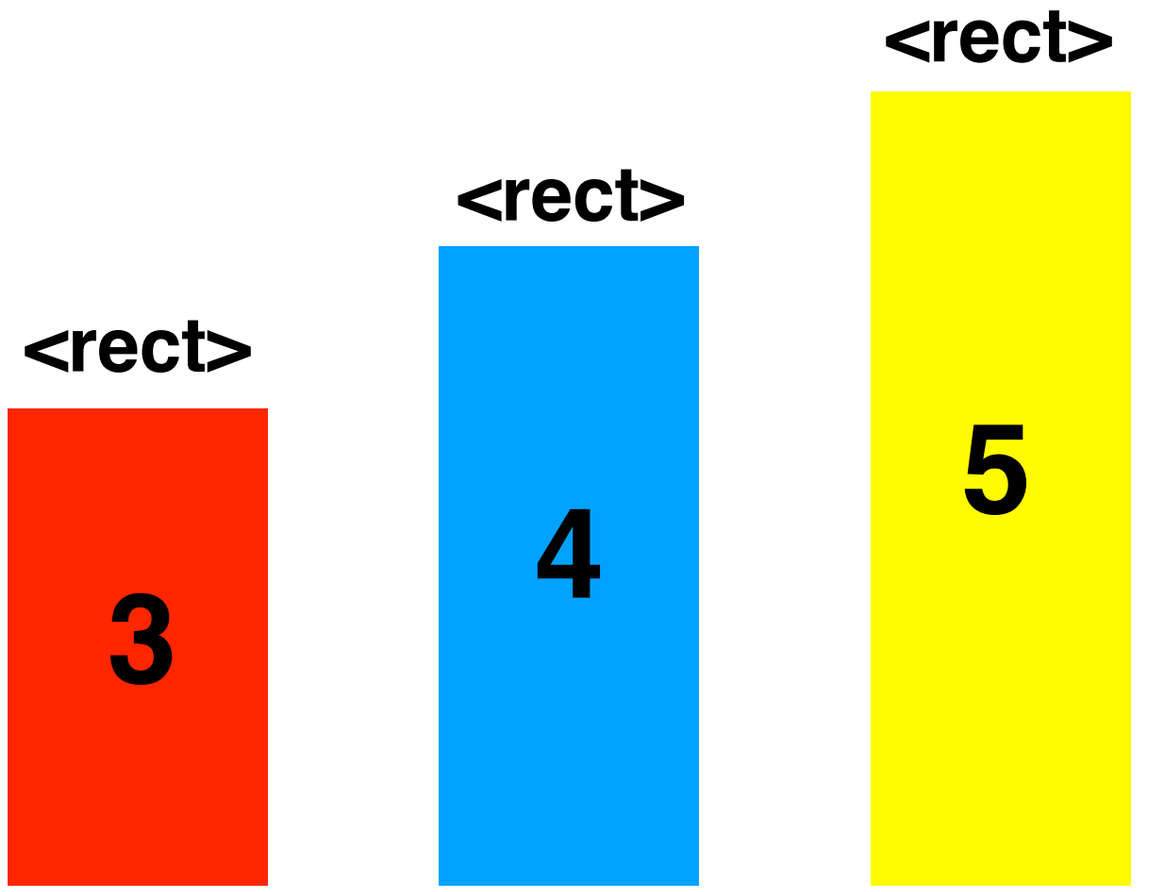
- bar 3 removed
- bar 1 switches to 2018, becomes blue
- bar 2 switches to 2019, becomes yellow

Why isn't object constancy maintained?



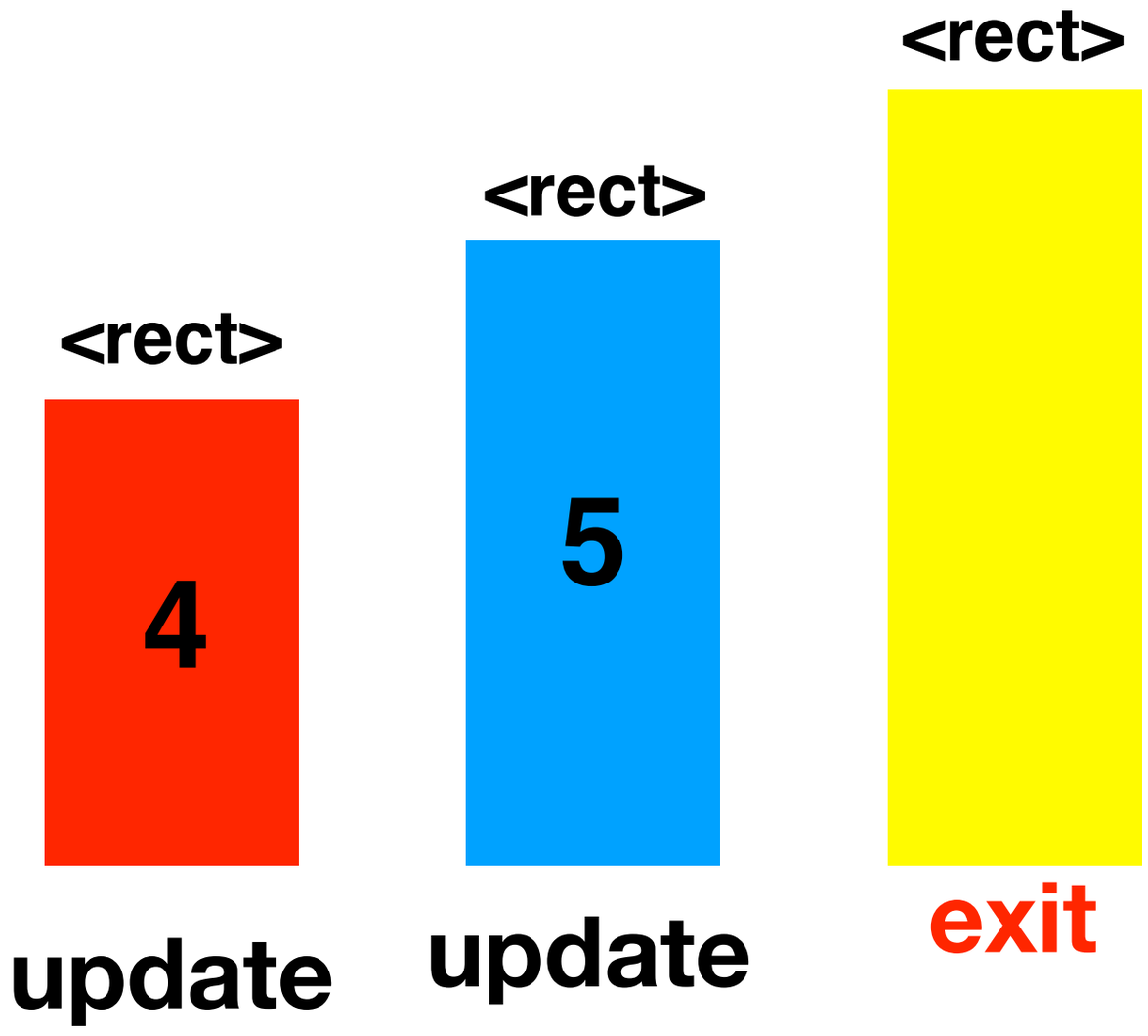
new data:
[4, 5]

What we *want* to happen



new data:
[4, 5]

What actually happens



new data:
[4, 5]

Solution: join data by key

not in order of SVG elements

ex.

```
const bardata = [
{key: 2017, value: 3},
{key: 2018, value: 4},
{key: 2019, value: 5}
];
```

Bind data by key

key: 2017
value: 3

key: 2018
value: 4

key: 2019
value: 5

```
d3.select("svg")
   .selectAll("rect")
   .data([{key: 2018, value: 4},
          {key: 2019, value: 5}],
          d => d.key);
```

What selections are returned?

Bind data by key

key: 2017 value: 3	key: 2018 value: 4	key: 2019 value: 5

```
d3.select("svg")
   .selectAll("rect")
   .data([{key: 2018, value: 4},
          {key: 2019, value: 5}],
          d => d.key);
```

What selections are returned?

key: 2017 value: 3	key: 2018 value: 4	key: 2019 value: 5
exit	**update**	**update**

Bind data by key

key: 2017
value: 3

key: 2018
value: 4

key: 2019
value: 5

```
d3.select("svg")
   .selectAll("rect")
   .data([{key: 2018, value: 4},
          {key: 2020, value: 7}],
          d => d.key);
```

What selections are returned?

Bind data by key

key: 2017
value: 3

key: 2018
value: 4

key: 2019
value: 5

```
d3.select("svg")
   .selectAll("rect")
   .data([{key: 2018, value: 4},
          {key: 2020, value: 7}],
          d => d.key);
```

What selections are returned?

key: 2017
value: 3

exit

key: 2018
value: 4

update

key: 2019
value: 5

exit

key: 2020
value: 7

enter

How to bind data by key

make data an array of objects:

ex.

```
const 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

```
> const 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}
```

How to bind data by key

- **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()