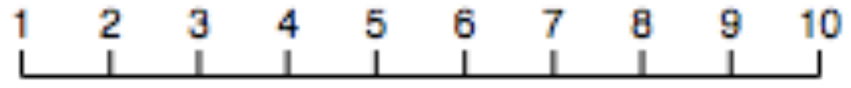
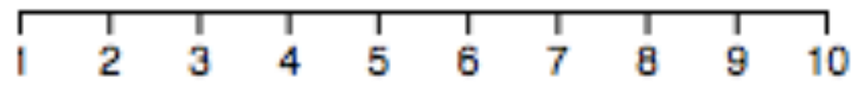
Axis Components

construct axis generators for given scales

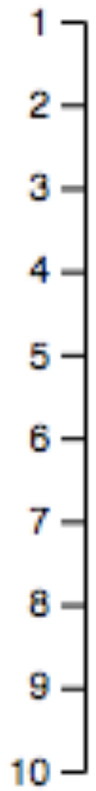
`d3.axisTop()`



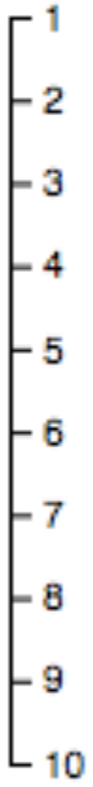
`d3.axisBottom()`



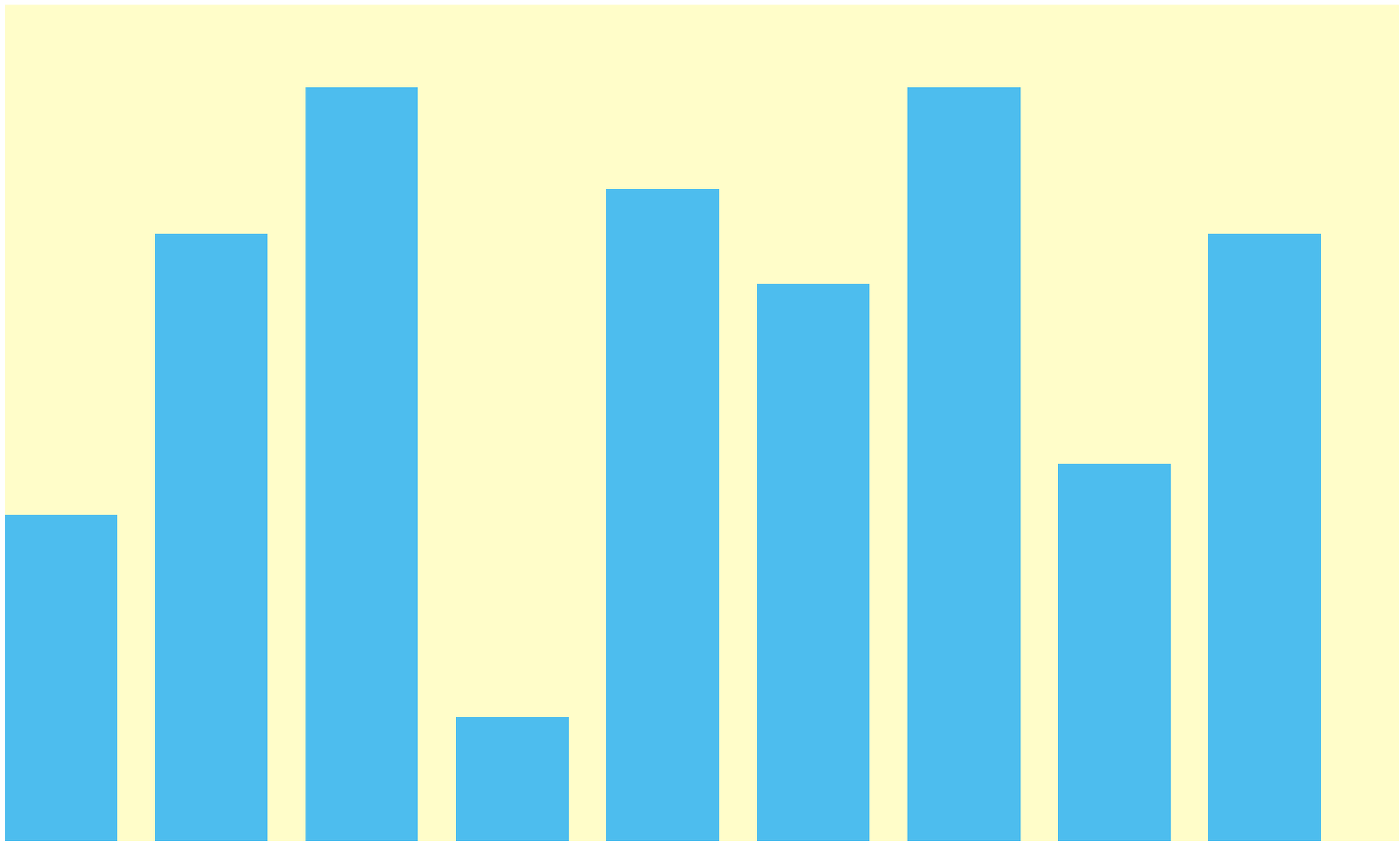
`d3.axisLeft()`



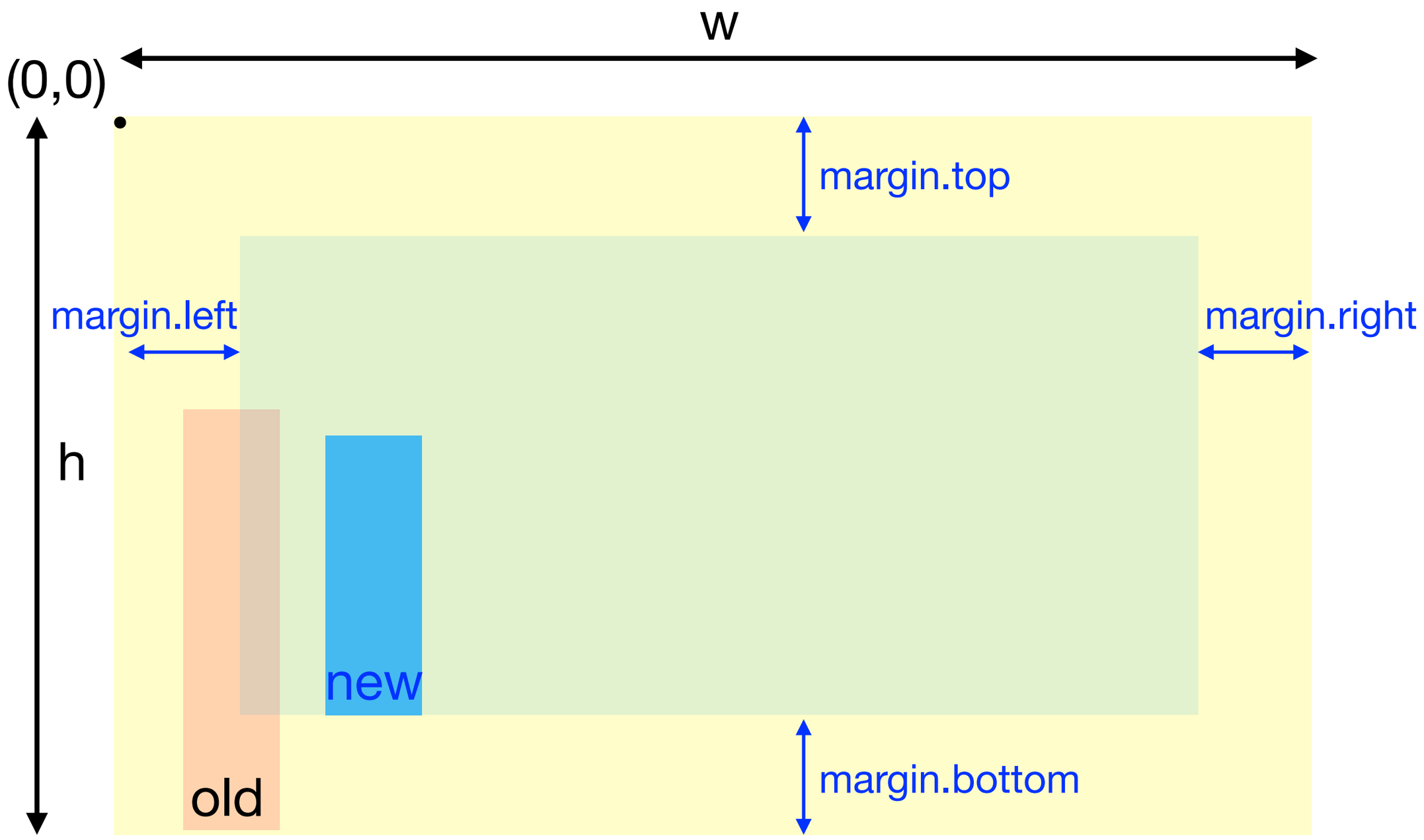
`d3.axisRight()`



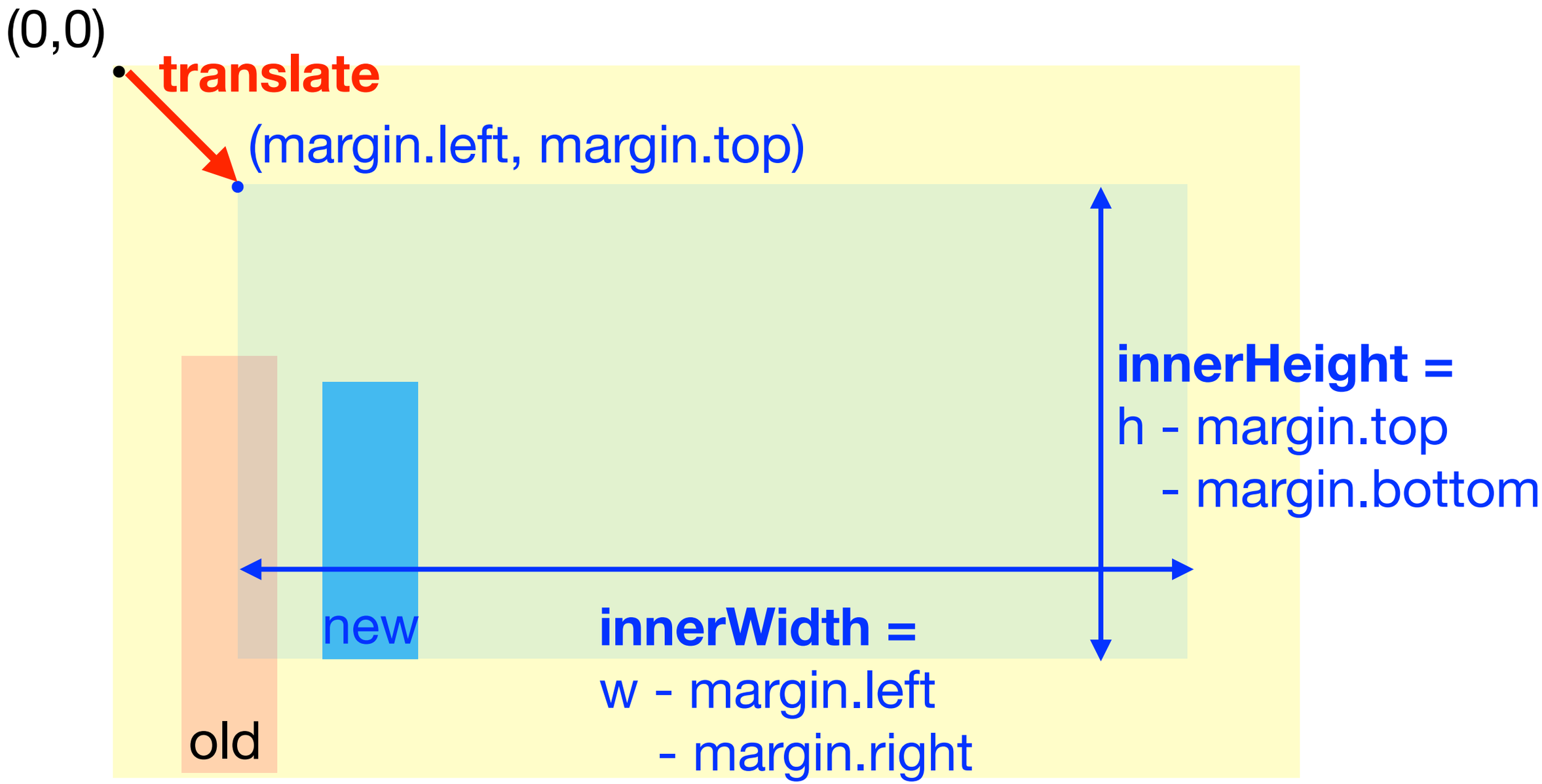
Axes need space



Margin convention



Margin convention



Add background rectangle

```
svg.append("rect")
      .attr("x", 0)
      .attr("y", 0)
      .attr("width", w)
      .attr("height", h)
      .attr("fill", "lightblue");
```

Groups

```
var bars = svg.append("g")
   .selectAll("rect").data(bardata)
   .enter().append("rect");
```

before

```
<svg>
   <rect> ... </rect>
   <rect> ... </rect>
   <rect> ... </rect>
</svg>
```

after

```
<svg>
   <g>
      <rect> ... </rect>
      <rect> ... </rect>
      <rect> ... </rect>
   </g>
</svg>
```

Groups

```
<svg>
  <rect> ... </rect>
  <g>
    <rect> ... </rect>
    <rect> ... </rect>
  </g>
</svg>
```

```
svg.select("rect")
```

```
svg.select("g")
   .selectAll("rect")
```

Groups

add an id

```
var bars = svg.append("g")
      .attr("id", "plot")
    .selectAll("rect")
      .data(bardata);
```

```
<g id="plot">
```

Groups

translate

```
var bars = svg.append("g")
     .attr("transform",
           `translate (${margin.left},
                       ${margin.top})`)
   .selectAll("rect")
     .data(bardata);
```

```
<g transform="translate (25, 25)">
```

ES6 template literals, use ``

```
var bars = svg.append("g")
    .attr("transform",
          `translate (${margin.left},
                      ${margin.top})`)
 .selectAll("rect")
   .data(bardata);
```

OR

```
"translate (" + margin.left + ","
    + margin.top + ")")
```

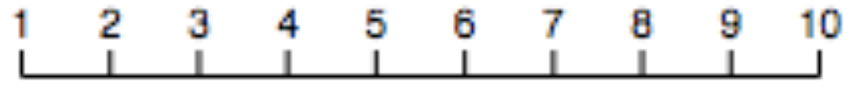
EDAV6_1_scaleLinear.html

margins

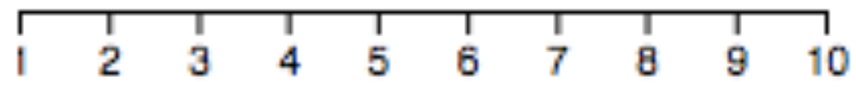
Axis Components

construct axis generators for given scales

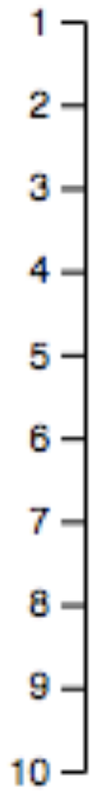
`d3.axisTop()`



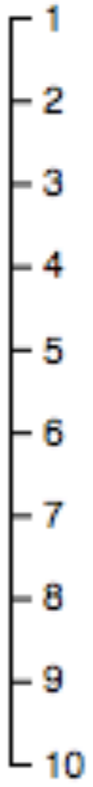
`d3.axisBottom()`



`d3.axisLeft()`

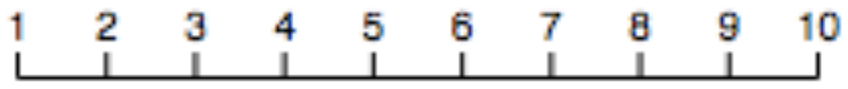


`d3.axisRight()`



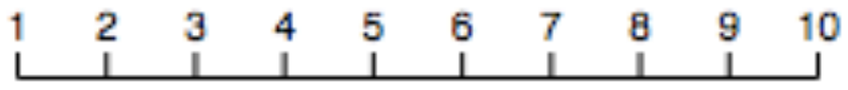
Axes

d3.axisBottom()



Axes

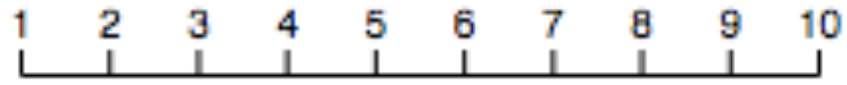
d3.axisBottom()



scale

```
var xScale = d3.scaleLinear()
    .domain([1,10])
    .range([0,200]);
```

Axes

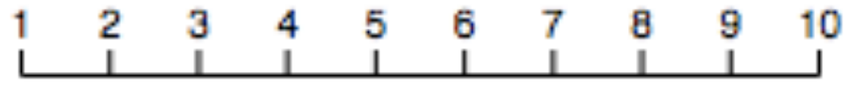


axis generator

```
var xAxis = d3.axisBottom()
    .scale(xScale);
```

or `var xAxis = d3.axisBottom(xScale);`

Axes



When called on a selection, the axis generator creates axis SVG elements

```
d3.select("svg").append("g")
    .call(xAxis);
```

or

```
xAxis(d3.select("svg").append("g"));
```

Generated SVG axis elements

```
<g fill="none" font-size="10" font-family="sans-serif" text-anchor="middle">

    <path class="domain" stroke="#000" d="M0.5,6V0.5H200.5V6"></path>

    <g class="tick" opacity="1" transform="translate(0.5,0)">
        <line stroke="#000" y2="6"></line>
        <text fill="#000" y="9" dy="0.71em">1</text>
    </g>

    <g class="tick" opacity="1" transform="translate(22.72222222222222,0)">
        <line stroke="#000" y2="6"></line>
        <text fill="#000" y="9" dy="0.71em">2</text>
    </g>

    (8 more tick mark / tick label groups)

</g>
```

Generated SVG axis elements

axis component

axis generator

d3.axisBottom(*scale*) --> xAxis

xAxis([*selection*]) --> SVG elements

Possible, but not advisable:

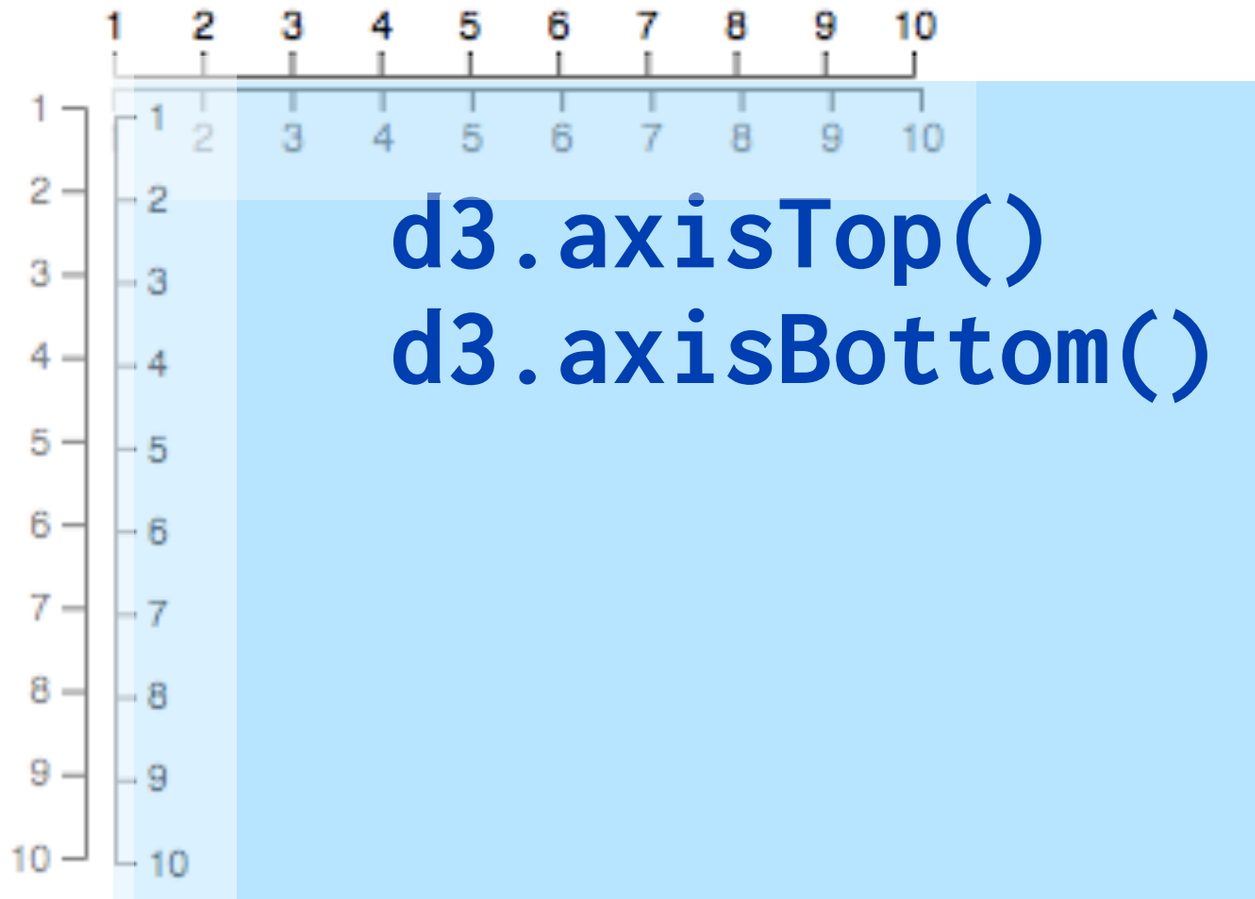
```
d3.axisBottom(d3.scaleLinear()
     .domain([1,10])
         .range([0,200]))(d3.select("svg")
                             .append("g"));
```

Axis Components

control orientation not location on the svg

all axes are rendered at the origin

`d3.axisLeft()`
`d3.axisRight()`



Translate axes to position them

```
var yAxis = d3.axisLeft()
    .scale(yScale);

svg.append("g")
    .attr("class", "yAxis")
    .attr("transform",
        `translate({$margin.left},
                   ${margin.top})`)
    .call(yAxis);
```

EDAV6_2_yaxis.html

Practice: add x-axis