

# Drawing with Data

---

VI Workshop am 17.05.2022

# Struktur

---

## 1. Input-Session

- div-Elemente und Bar-Charts
- svg-Elemente und Bar-Charts
- Labels
- Kreise
- Scatterplot

## 2. Workshop-Task

**<div> vs. <svg>**

# <div> </div>

- Element des HTML-DOM-Tree
- Hat keine semantische Bedeutung
- Gruppierung von Elementen allein zum Zweck der Gestaltung
- Kann im CSS über die Attribute `class()` oder `id()` verändert werden

# Bars

---

HTML:

```
<div class="bar"></div>
```

CSS:

```
div.bar {  
  display: inline-block;  
  width: 20px;  
  height: 75px;  
  background-color: teal;  
}
```

# Bars

---



# Bar-Charts

---

JavaScript:

```
var dataset = [25, 7, 5, 26, 11, 8, 25, 14,  
23, 19, 14, 11, 22, 29, 11, 13, 12, 17, 18,  
10, 24, 18, 25, 9, 3];
```

```
D3.select(„body“).selectAll(„div“)  
  .data(dataset)  
  .enter()  
  .append(„div“)  
  .attr(„class“, „bar“)  
  .style("height", function(d) {  
    var barHeight = d * 5;  
    return barHeight + "px";  
  });
```

# Bar-Charts

---

JavaScript:

```
var dataset = [25, 7, 5, 26, 11, 8, 25, 14,  
23, 19, 14, 11, 22, 29, 11, 13, 12, 17, 18,  
10, 24, 18, 25, 9, 3];
```

```
D3.select(„body“).selectAll(„div“)  
  .data(dataset)  
  .enter()  
  .append(„div“)  
  .attr(„class“, „bar“)  
  .style("height", function(d) {  
    var barHeight = d * 5;  
    return barHeight + "px";  
  });
```



# Bar-Charts

---

JavaScript:

```
var dataset = [25, 7, 5, 26, 11, 8, 25, 14,  
23, 19, 14, 11, 22, 29, 11, 13, 12, 17, 18,  
10, 24, 18, 25, 9, 3];
```

```
D3.select(„body“).selectAll(„div“)  
  .data(dataset)  
  .enter()  
  .append(„div“)  
  .attr(„class“, „bar“)  
  .style(„height“, function(d) {  
    var barHeight = d * 5;  
    return barHeight + „px“;  
  });
```

# Bar-Charts

---

JavaScript:

```
var dataset = [25, 7, 5, 26, 11, 8, 25, 14,  
23, 19, 14, 11, 22, 29, 11, 13, 12, 17, 18,  
10, 24, 18, 25, 9, 3];
```

```
D3.select(„body“).selectAll(„div“)  
  .data(dataset)  
  .enter()  
  .append(„div“)  
  .attr(„class“, „bar“)  
  .style("height", function(d) {  
    var barHeight = d * 5;  
    return barHeight + "px";  
  });
```

# Bar-Charts

---

JavaScript:

```
var dataset = [25, 7, 5, 26, 11, 8, 25, 14,  
23, 19, 14, 11, 22, 29, 11, 13, 12, 17, 18,  
10, 24, 18, 25, 9, 3];
```

```
D3.select(„body“).selectAll(„div“)  
  .data(dataset)  
  .enter()  
  .append(„div“)  
  .attr(„class“, „bar“)  
  .style("height", function(d) {  
    var barHeight = d * 5;  
    return barHeight + "px";  
  });
```

# Bar-Charts

---

JavaScript:

```
var dataset = [25, 7, 5, 26, 11, 8, 25, 14,  
23, 19, 14, 11, 22, 29, 11, 13, 12, 17, 18,  
10, 24, 18, 25, 9, 3];
```

```
D3.select(„body“).selectAll(„div“)  
  .data(dataset)  
  .enter()  
  .append(„div“)  
  .attr(„class“, „bar“)  
  .style("height", function(d) {  
    var barHeight = d * 5;  
    return barHeight + "px";  
  });
```

# Bar-Charts

---

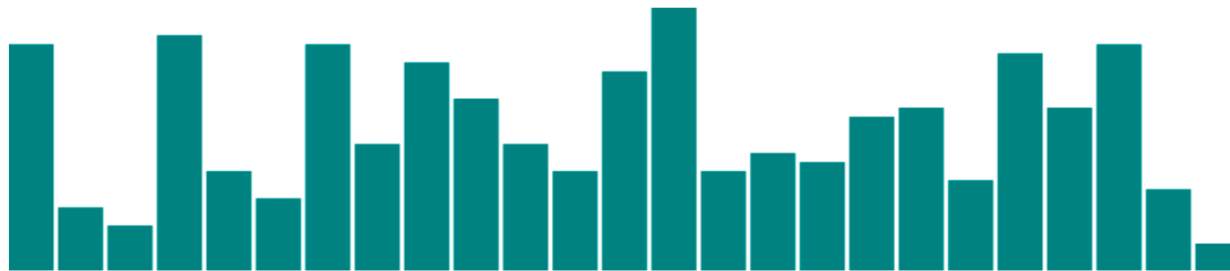
JavaScript:

```
var dataset = [25, 7, 5, 26, 11, 8, 25, 14,  
23, 19, 14, 11, 22, 29, 11, 13, 12, 17, 18,  
10, 24, 18, 25, 9, 3];
```

```
D3.select(„body“).selectAll(„div“)  
  .data(dataset)  
  .enter()  
  .append(„div“)  
  .attr(„class“, „bar“)  
  .style("height", function(d) {  
    var barHeight = d * 5;  
    return barHeight + "px";  
  });
```

# Bar-Charts

---



# <svg> </svg>

- „*Scalable Vector Graphics*“
- SVG Grafiken lassen sich unbegrenzt skalieren
- Bietet unterschiedliche Formen (Linien, Rechtecke, Kreise, Ellipsen, ...)
- Alle Eigenschaften sind als Attribute spezifiziert
  - `append()` and `attr()` möglich

→ Grundsätzlich mehr Flexibilität

# Bars in SVG

---

## Creating SVG

```
var w = 500;  
var h = 100;  
  
var dataset=[5,10,13,19,21,25,22,18,15,13,11,12,  
              15,20,18,17,16,18,23,25];  
  
    var svg = d3.select("body")  
                .append("svg")  
                .attr("width", w)  
                .attr("height", h);
```



# Bars in SVG

---

```
var w = 500;
var h = 100;

var dataset=[5,10,13,19,21,25,22,18,15,13,11,12,
              15,20,18,17,16,18,23,25];

var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);
```

# Bars in SVG

---

## Making a Bar Chart

```
svg.selectAll(„rect“)
  .data(dataset)
  .enter()
  .append(„rect“)
  .attr(„x“, 0)
  .attr(„y“, 0)
  .attr(„width“, 20)
  .attr(„height“, 100);
```

# Bars in SVG

---

```
svg.selectAll(„rect“)
  .data(dataset)
  .enter()
  .append(„rect“)
  .attr(„x“, 0)
  .attr(„y“, 0)
  .attr(„width“, 20)
  .attr(„height“, 100);
```

# Bars in SVG

---



# Bars in SVG

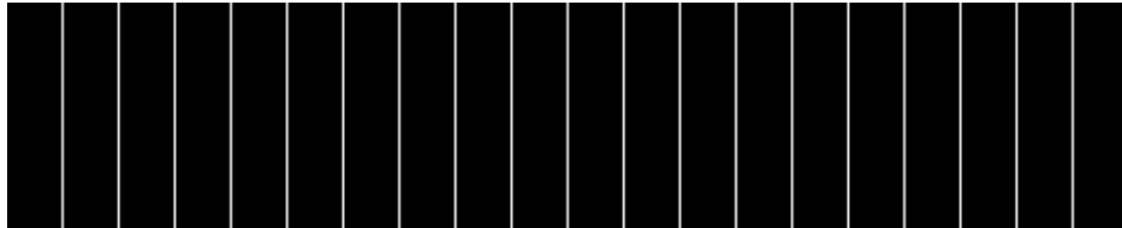
---

```
svg.selectAll(„rect“)
  .data(dataset)
  .enter()
  .append(„rect“)
  .attr(„x“, function(d, i) {
    return i * 21;
  })
  .attr(„y“, 0)
  .attr(„width“, 20)
  .attr(„height“, 100);
```

# Bars in SVG

---

Ergebnis bei 20 Daten im Datensatz



# Bars in SVG

---

```
svg.selectAll(„rect“)
  .data(dataset)
  .enter()
  .append(„rect“)
  .attr(„x“, function(d, i) {
    return i * (w / dataset.length);
  })
  .attr(„y“, 0)
  .attr(„width“, 20)
  .attr(„height“, 100);
```

# Bars in SVG

---





# Bars in SVG

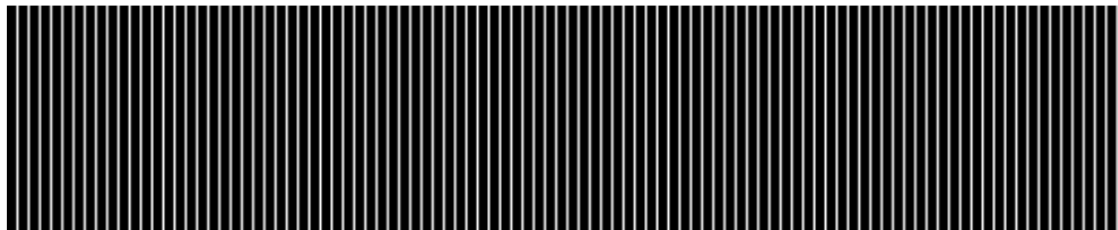
---

```
var w = 500;
var h = 100;
var barPadding = 1;

svg.selectAll(„rect“)
  .data(dataset)
  .enter()
  .append(„rect“)
  .attr(„x“, function(d, i) {
    return i * (w / dataset.length);
  })
  .attr(„y“, 0)
  .attr(„width“, w / dataset.length - barPadding)
  .attr(„height“, 100);
```

# Bars in SVG

---



# Bars in SVG

---

## Proportionale Bar-Breite durch Padding

```
svg.selectAll(„rect“)
  .data(dataset)
  .enter()
  .append(„rect“)
  .attr(„x“, function(d, i) {
    return i * (w / dataset.length);
  })
  .attr(„y“, 0)
  .attr(„width“, w / dataset.length - barPadding)
  .attr(„height“, function(d) {
    return d * 4;
  });
```

# Bar-Charts

---



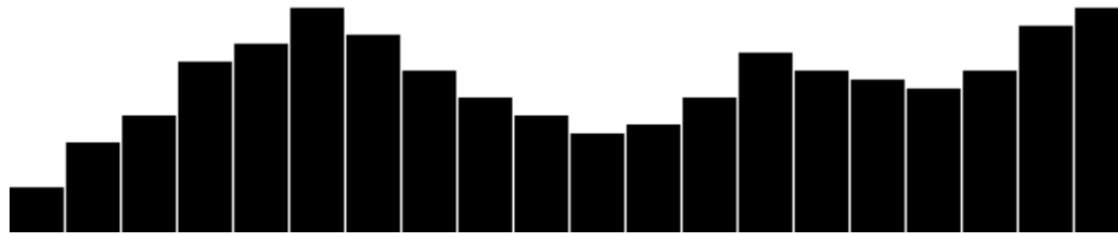
# Bars in SVG

---

```
svg.selectAll(„rect“)
  .data(dataset)
  .enter()
  .appended(„rect“)
  .attr(„x“, function(d, i) {
    return i * (w / dataset.length);
  })
  .attr(„y“, function(d) {
    return h - (d * 4)
  })
  .attr(„width“, w / dataset.length - barPadding)
  .attr(„height“, function(d) {
    return d * 4;
  });
```

# Bar-Charts

---



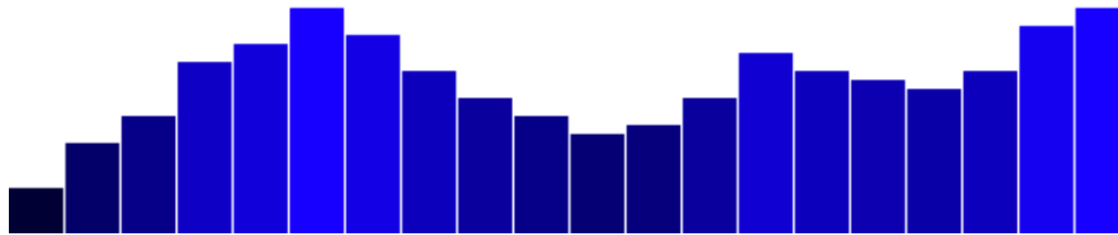
# Bar-Charts

---

```
svg.selectAll(„rect“)
  .data(dataset)
  .enter()
  .append(„rect“)
  .attr(„x“, function(d, i) {
    return i * (w / dataset.length);
  })
  .attr(„y“, function(d) {
    return h - (d * 4)
  })
  .attr(„width“, w / dataset.length - barPadding)
  .attr(„height“, function(d) {
    return d * 4;
  })
  .attr(„fill“, function(d) {
    return „rgb(0,0,“ + Math.round(d * 10) + „)“;
  });
```

# Bar-Charts

---





# Label

---

```
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function (d) {
    return d;
  })
  .attr("x", function (d, i) {
    return i * (w / dataset.length) + 5;
  })
  .attr("y", function (d) {
    return h - (d * 4) + 15;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", "11px")
  .attr("fill", "white");
```

# Label

---

```
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function (d) {
    return d;
  })
  .attr("x", function (d, i) {
    return i * (w / dataset.length) + 5;
  })
  .attr("y", function (d) {
    return h - (d * 4) + 15;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", "11px")
  .attr("fill", "white");
```

# Label

---

```
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function (d) {
    return d;
  })
  .attr("x", function (d, i) {
    return i * (w / dataset.length) + 5;
  })
  .attr("y", function (d) {
    return h - (d * 4) + 15;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", "11px")
  .attr("fill", "white");
```

# Label

---

```
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function (d) {
    return d;
  })
  .attr("x", function (d, i) {
    return i * (w / dataset.length) + 5;
  })
  .attr("y", function (d) {
    return h - (d * 4) + 15;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", "11px")
  .attr("fill", "white");
```

# Label

---

```
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function (d) {
    return d;
  })
  .attr("x", function (d, i) {
    return i * (w / dataset.length) + 5;
  })
  .attr("y", function (d) {
    return h - (d * 4) + 15;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", "11px")
  .attr("fill", "white");
```

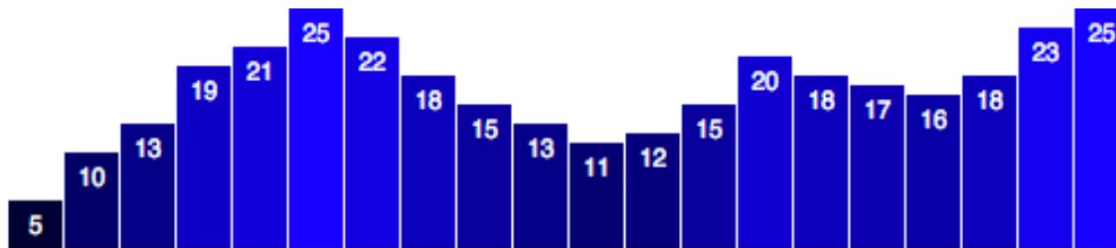
# Label

---

```
svg.selectAll("text")
  .data(dataset)
  .enter()
  .append("text")
  .text(function (d) {
    return d;
  })
  .attr("x", function (d, i) {
    return i * (w / dataset.length) + 5;
  })
  .attr("y", function (d) {
    return h - (d * 4) + 15;
  })
  .attr("font-family", "sans-serif")
  .attr("font-size", "11px")
  .attr("fill", "white");
```

# Label

---



# Kreise

---

```
var w = 500;  
var h = 50;  
var dataset = [ 5, 10, 15, 20, 25 ];
```

```
var svg = d3.select("body")  
  .append("svg")  
  .attr("width", w)  
  .attr("height", h);
```



# Kreise

---

```
var circles = svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle");

circles.attr("cx", function(d, i) {
    return (i * 50) + 25;
})
    .attr("cy", h/2)
    .attr("r", function(d) {
        return d;
    });
```

# Kreise

---

```
var circles = svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle");

circles.attr("cx", function(d, i) {
    return (i * 50) + 25;
})
    .attr("cy", h/2)
    .attr("r", function(d) {
        return d;
    });
```

# Kreise

---

```
var circles = svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle");

circles.attr("cx", function(d, i) {
    return (i * 50) + 25;
})
    .attr("cy", h/2)
    .attr("r", function(d) {
        return d;
    });
```

# Kreise

---

```
var circles = svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle");

circles.attr("cx", function(d, i) {
    return (i * 50) + 25;
})
.attr("cy", h/2)
.attr("r", function(d) {
    return d;
});
```

# Kreise

---

```
var circles = svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle");

circles.attr("cx", function(d, i) {
    return (i * 50) + 25;
})
    .attr("cy", h/2)
    .attr("r", function(d) {
        return d;
    });
```

# Kreise

---



# Kreise

---

```
.attr("fill", "yellow")  
.attr("stroke", "orange")  
.attr("stroke-width", function (d) {  
    return d / 2;  
});
```



# Scatterplot

---

```
var dataset = [  
    [ 5, 20 ],  
    [ 480, 90 ],  
    [ 250, 50 ],  
    [ 100, 33 ],  
    [ 330, 95 ],  
    [ 410, 12 ],  
    [ 475, 44 ],  
    [ 25, 67 ],  
    [ 85, 21 ],  
    [ 220, 88 ],  
];
```



# Scatterplot

---

```
var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);

svg.selectAll(„circle“)
    .data(dataset)
    .enter()
    .append(„circle“)
    .attr(„cx“, function(d) {
        return d[0];
    })
    .attr(„cy“, function(d) {
        return d[1];
    })
    .attr(„r“, 5);
```

# Scatterplot

---

```
var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);

svg.selectAll(„circle“)
    .data(dataset)
    .enter()
    .append(„circle“)
    .attr(„cx“, function(d) {
        return d[0];
    })
    .attr(„cy“, function(d) {
        return d[1];
    })
    .attr(„r“, 5);
```

# Scatterplot

---



# Workshop Task

# Setup

---

GitHub-Repository mit Setup: <https://github.com/leo-3108/vi-workshop-drawing-with-data>

- Forken
- Clonen
- Mit lokalem Server öffnen

# Aufgabe

---

Erstelle ein Bar-Chart, das den vorgegebenen Temperaturdatensatz abbildet. Folgende Anforderungen sollen berücksichtigt werden:

1. Die Temperatur wird durch die Höhe der Bars abgebildet
2. Farbliche Gestaltung der einzelnen Bars:
  - a) Temperatur  $\geq 15$  Grad: Rot
  - b) Temperatur  $< 15$  Grad: Blau
3. Beschrifte die einzelnen Bars mit den jeweiligen Monatsnamen

Bearbeitungszeit: 15-20 Minuten  
Danach Besprechung