

INTERAKTIVE DATENVISUALISIERUNG IM WEB MIT D3.js

EINLEITUNG

DATENVISUALISIERUNG IM WEB

Ziel

Leichter und verständlicher Zugang der Informationen für den Betrachter

Interdisziplinäres Aufgabengebiet

Erfordert Kenntnisse aus verschiedenen Disziplinen, wie z.B. der Mathematik und Statistik, der Gestaltungslehre und Psychologie sowie der Informatik

Gleichzeitiger Einsatz von verschiedenen Werkzeugen und Webtechnologien

HTML (Hypertext Markup Language) für den Seiteninhalt

CSS (Cascading Style Sheets) für die visuelle Gestaltung

JavaScript für die Interaktion

SVG (Scalable Vector Graphics) für die eigentlichen Grafiken

D3 (DATA-DRIVEN DOCUMENTS)

D3.js vereinfacht das Zusammenspiel zwischen den einzelnen Webtechnologien (HTML, CSS, JS, SVG).

Die JavaScript-Bibliothek ermöglicht die Manipulation des Document Object Model (DOM) von HTML-Dokumenten auf der Grundlage von Daten.

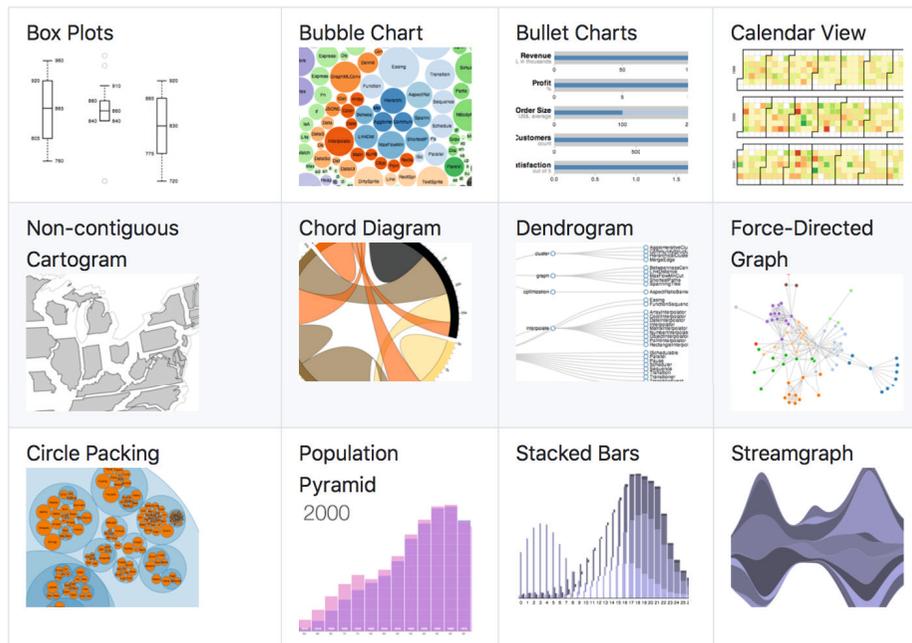
Im Gegensatz zu anderen Werkzeugen bietet D3.js **keine** vordefinierten Visualisierungen.

ANWENDUNGSMÖGLICHKEITEN

Wiki > Gallery

Welcome to the **D3 gallery**! More examples are available for forking on [Observable](#); see [D3's profile](#) and the [visualization collection](#). Please share your work on Observable, or [tweet us a link](#)!

Visual Index



Pages 62



Data-Driven Documents

- [Home](#)
- [Gallery](#)
- [Examples](#)
- [Tutorials](#)
- [Plugins](#)

Help

- [Stack Overflow](#)
- [Slack](#)
- [Google Group](#)
- [Gitter](#)

<https://github.com/d3/d3/wiki/Gallery>

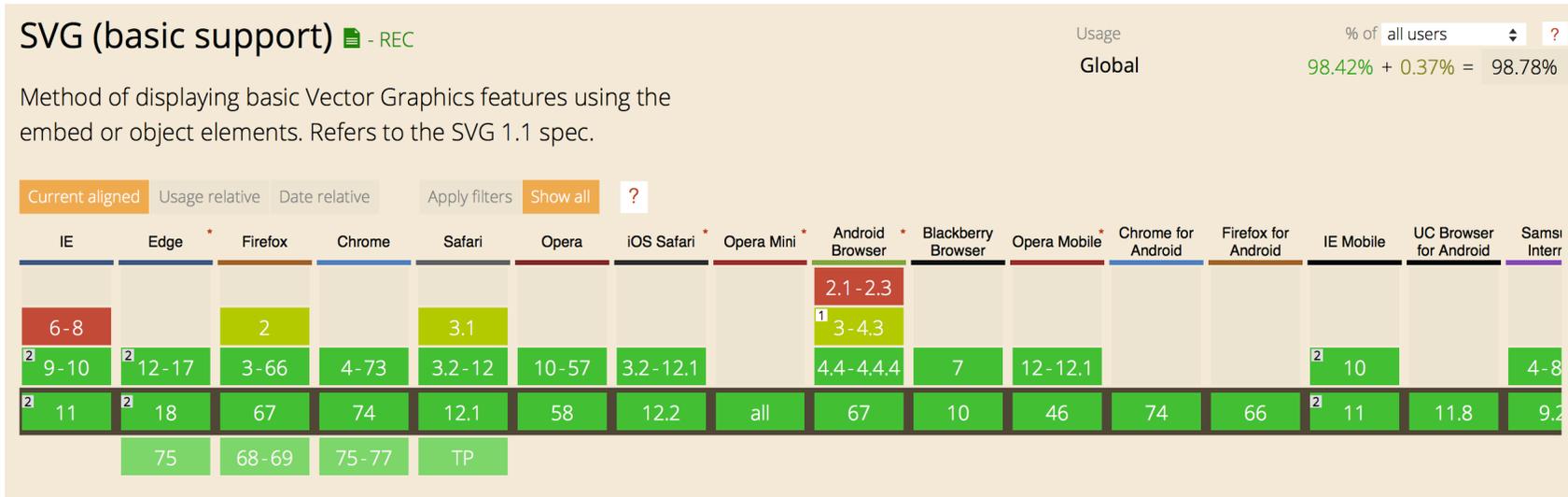
EXKURS: EINSATZ VON SVG IM WEB

SVG – VORTEILE

SVG-Grafiken verlieren niemals an Qualität.

Im Gegensatz zu Grafiken im Format JPEG oder PNG können SVG-Grafiken mit einem einfachen Texteditor erzeugt und bearbeitet werden.

SVG – KOMPATIBILITÄT



<https://caniuse.com/#feat=svg>

SVG ALS GRAFIKREFERENZ EINBINDEN

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Einbinden einer SVG-Grafik</title>
  </head>
  <body>
    <h1>SVG als Grafikreferenz</h1>
    <p>
      <br>
      <br>
      
    </p>
  </body>
</html>
```

SVG DIREKT IN DIE WEBSITE EINBETTEN

```
<h1>SVG direkt einbetten</h1>

<svg width="400" height="60">

  <!-- Rechteck -->
  <rect x="0" y="0" width="50" height="50" fill="green"></rect>

  <!-- Kreis -->
  <circle cx="90" cy="25" r="25" fill="red"></circle>

  <!-- Ellipse -->
  <ellipse cx="145" cy="25" rx="15" ry="25" fill="grey"></ellipse>

  <!-- Linie -->
  <line x1="185" y1="5" x2="230" y2="40" stroke="blue" stroke-width="5"></line>

  <!-- Text -->
  <text x="260" y="25" font-size="20px" fill="orange">Hallo Welt</text>

</svg>
```

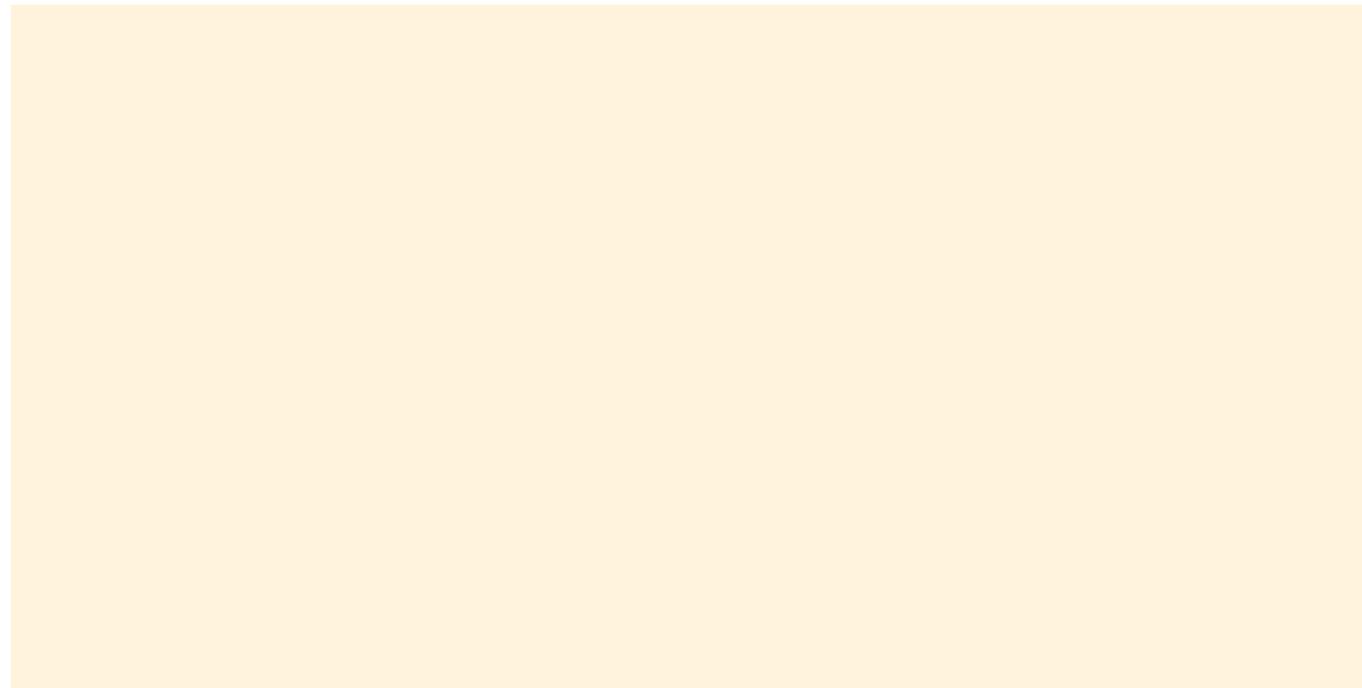
DIE GRAFISCHEN ELEMENTE VON SVG

```
<svg width="400" height="60"></svg>
```

KOORDINATENSYSTEM

$(0,0)$

$(400,0)$



$(0,60)$

$(400,60)$

KOORDINATENSYSTEM



DIE GRAFISCHEN ELEMENTE VON SVG – RECT

```
<svg width="400" height="60">  
  <rect x="0" y="0" width="50" height="50" fill="green"></rect>  
</svg>
```

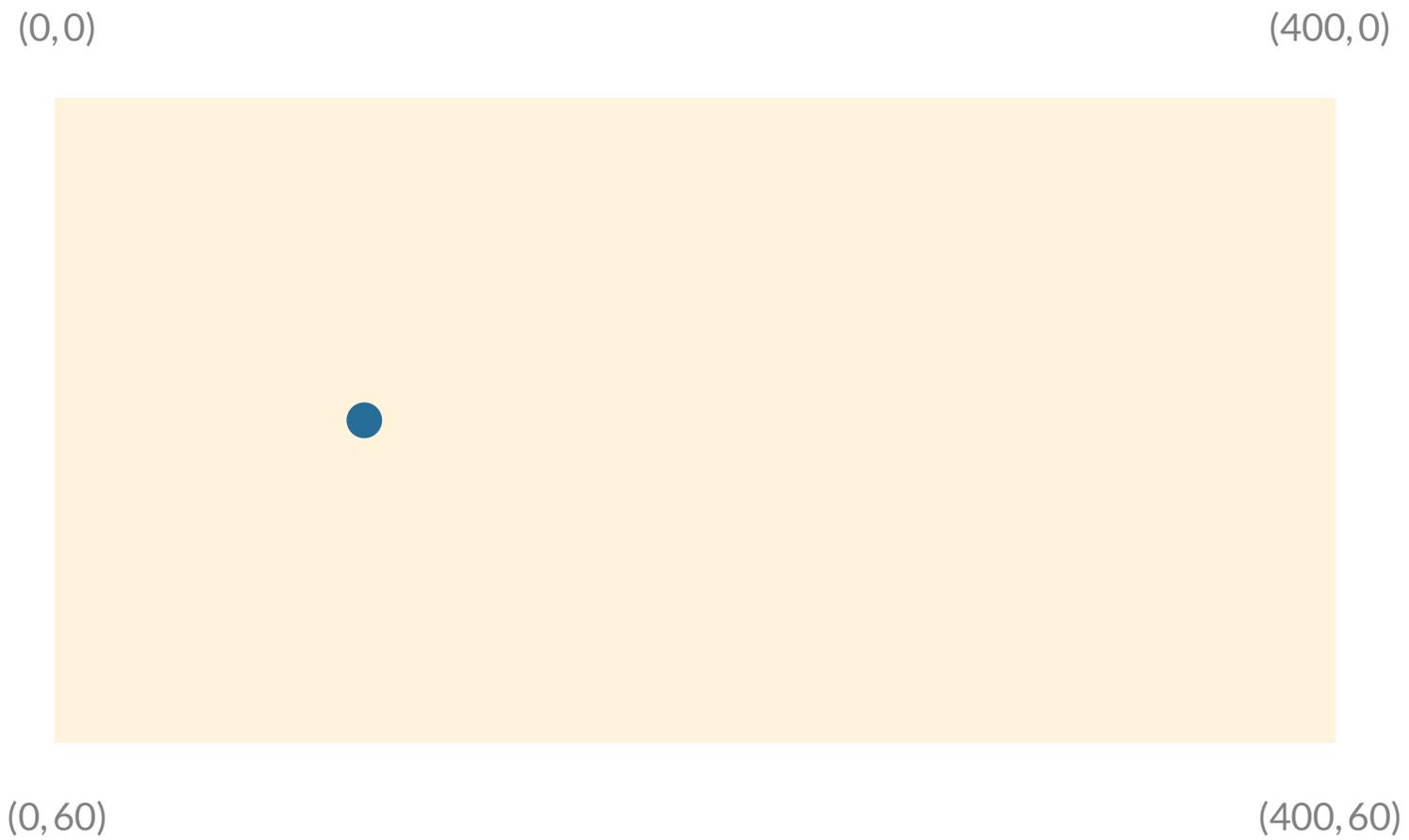
KOORDINATENSYSTEM – RECT



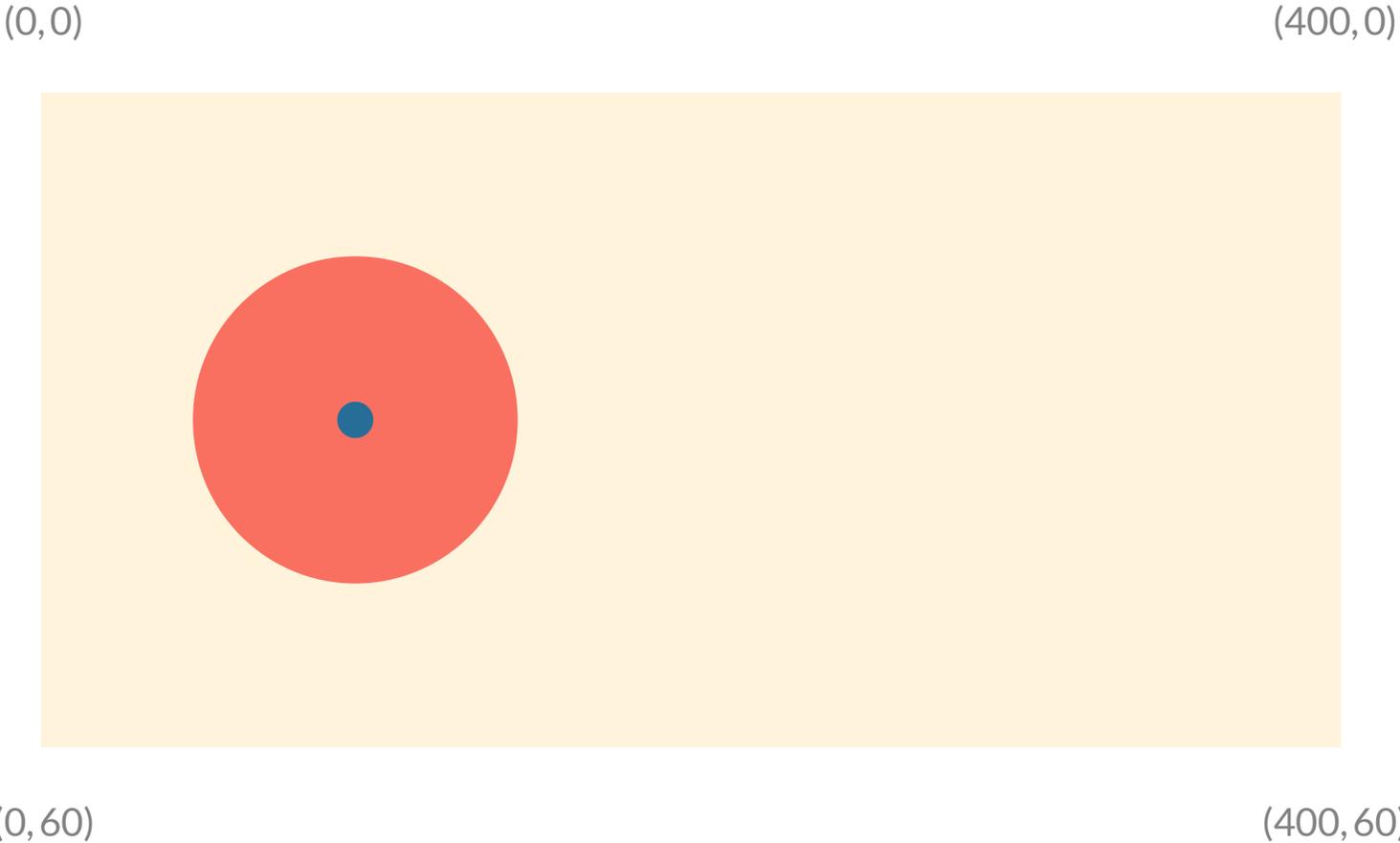
DIE GRAFISCHEN ELEMENTE VON SVG – CIRCLE

```
<svg width="400" height="60">  
  <circle cx="90" cy="25" r="25" fill="red"></circle>  
</svg>
```

KOORDINATENSYSTEM – CIRCLE



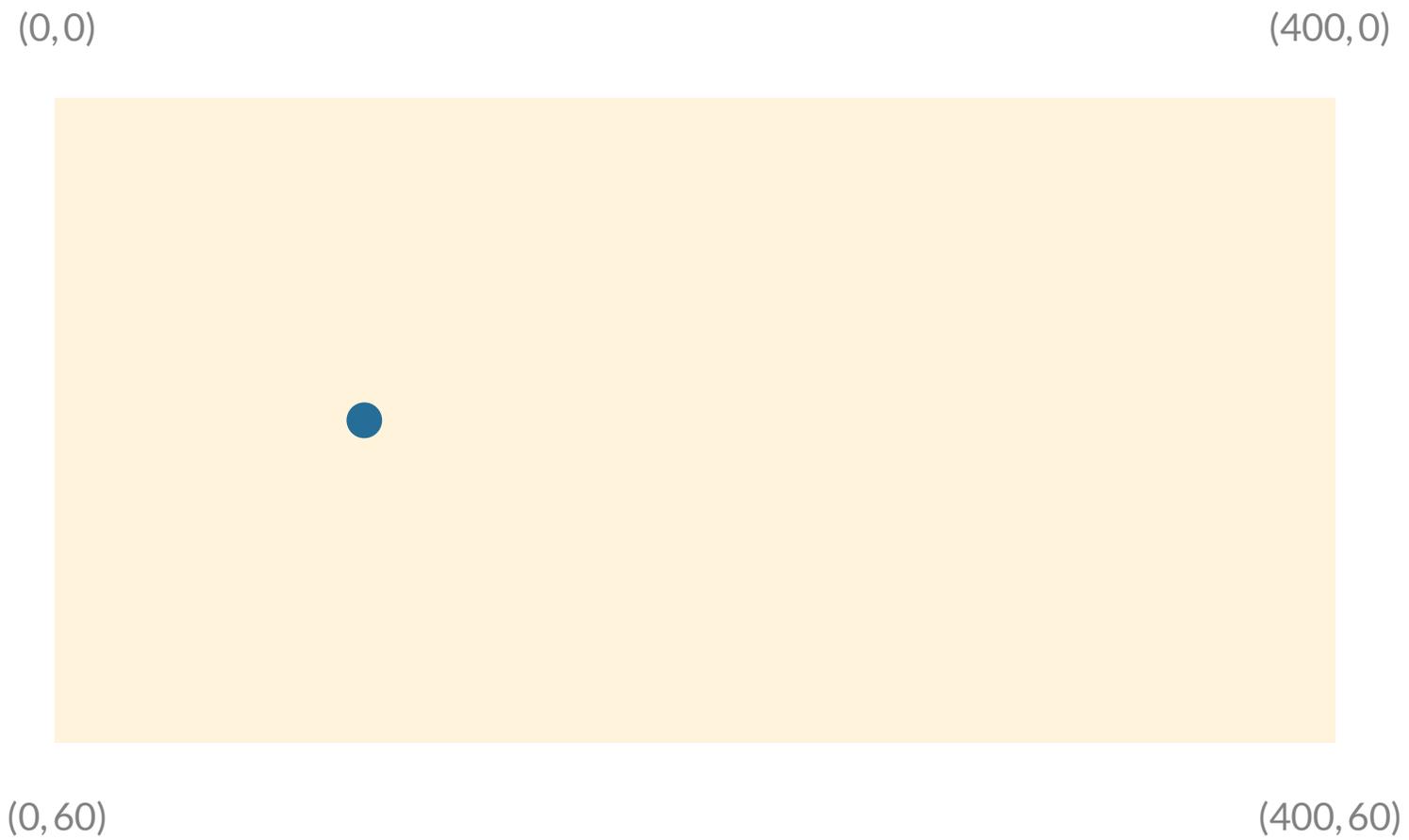
KOORDINATENSYSTEM – CIRCLE



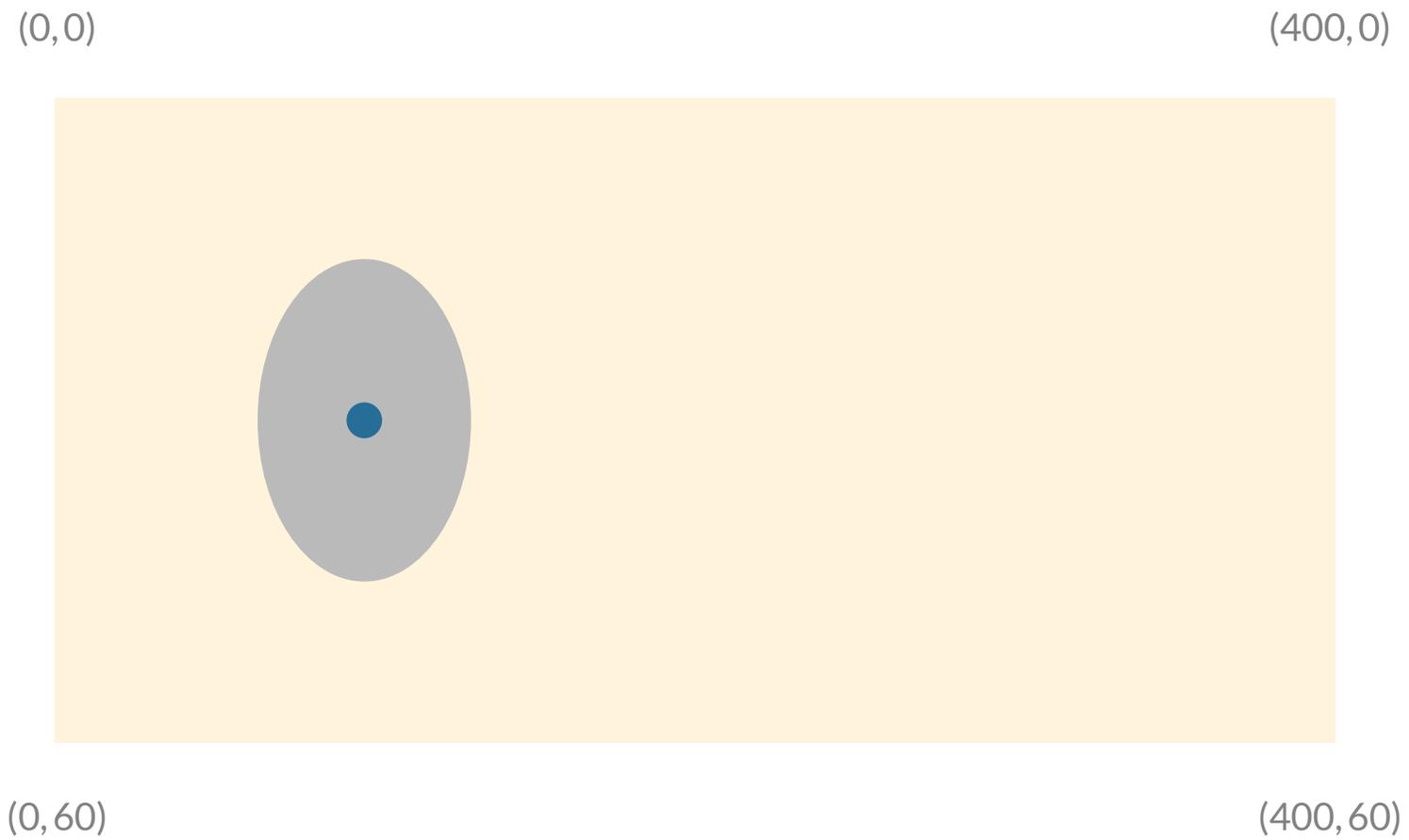
DIE GRAFISCHEN ELEMENTE VON SVG – ELLIPSE

```
<svg width="400" height="60">  
  <ellipse cx="145" cy="25" rx="15" ry="25" fill="grey"></ellipse>  
</svg>
```

KOORDINATENSYSTEM – ELLIPSE



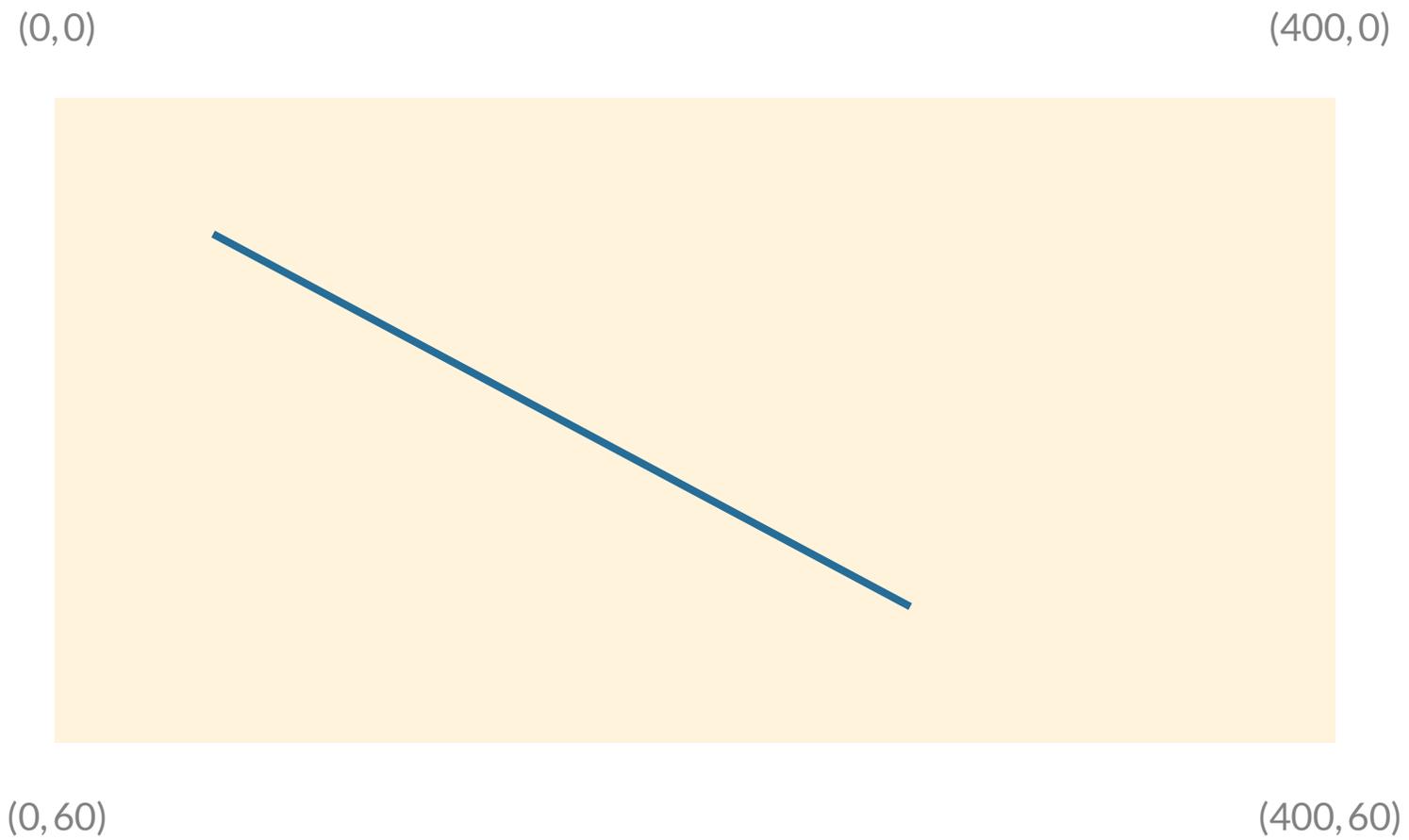
KOORDINATENSYSTEM – ELLIPSE



DIE GRAFISCHEN ELEMENTE VON SVG – LINE

```
<svg width="400" height="60">  
  <line x1="185" y1="5" x2="230" y2="40" stroke="blue" stroke-width="5"></line>  
</svg>
```

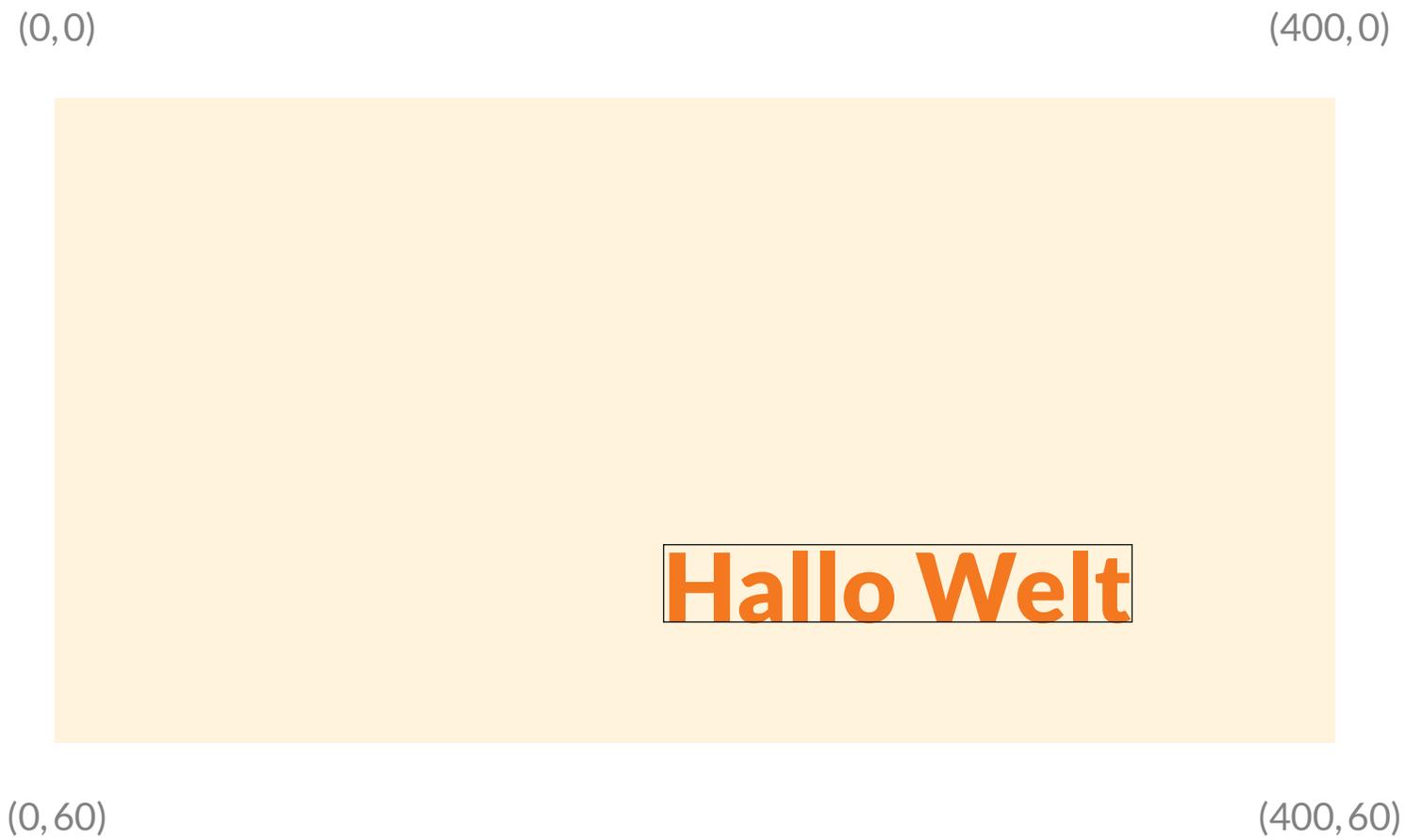
KOORDINATENSYSTEM – LINE



DIE GRAFISCHEN ELEMENTE VON SVG – TEXT

```
<svg width="400" height="60">  
  <text x="260" y="25" font-size="20px" fill="orange">Hallo Welt</text>  
</svg>
```

KOORDINATENSYSTEM – TEXT



FUNKTIONSWEISE VON D3.js

D3 SELECT

HTML/
SVG

```
<svg width="400" height="60">  
  <rect class="outside" x="0" y="0" width="50" height="50" fill="green"></rect>  
  <rect id="center" x="60" y="0" width="50" height="50" fill="green"></rect>  
  <rect class="outside" x="120" y="0" width="50" height="50" fill="green"></rect>  
</svg>
```

JavaScript

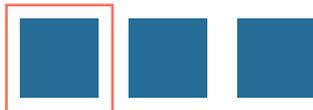
`d3.select("rect")`

`d3.select("#center")`

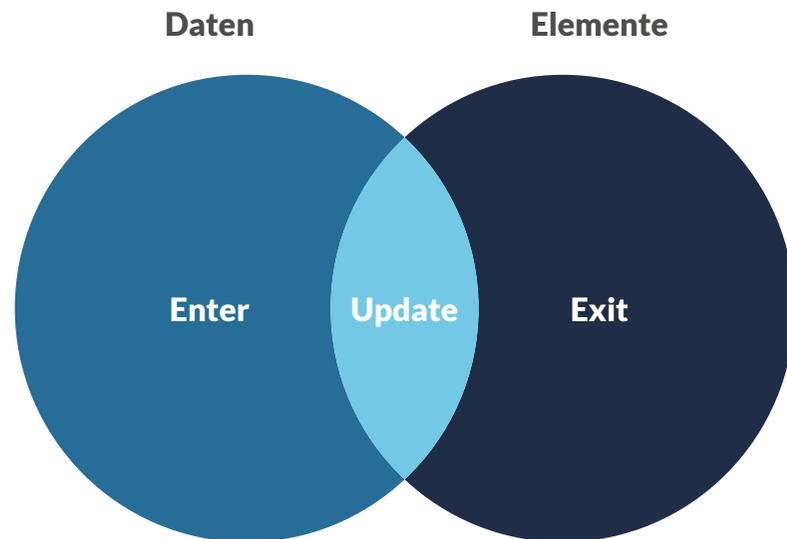
`d3.selectAll("#rect")`

`d3.select(".outside")`

Browser



D3 DATA JOIN



Möglichkeiten

Fall 1: Enter

Anzahl der Elemente < Anzahl der Datenpunkte

Fall 2: Update

Anzahl der Elemente = Anzahl der Datenpunkte

Fall 3: Exit

Anzahl der Elemente > Anzahl der Datenpunkte

D3 UPDATE PATTERN

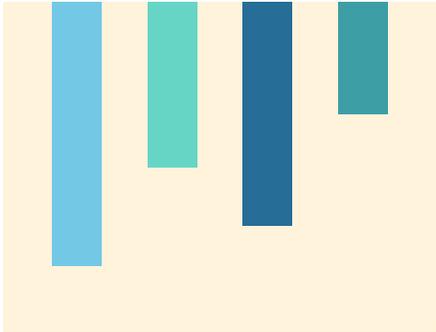
1. **Data Join**
Auswahl aller Elemente mit `d3.selectAll()` und Aktualisierung der Daten
2. **Exit**
Entfernen aller Elemente mit `exit()`, die in dem neuen Datensatz nicht existieren
3. **Update**
Setzen der Attribute für die existierenden Elemente
4. **Enter**
Erstellung neuer Elemente mit `enter()`

D3 SCALES

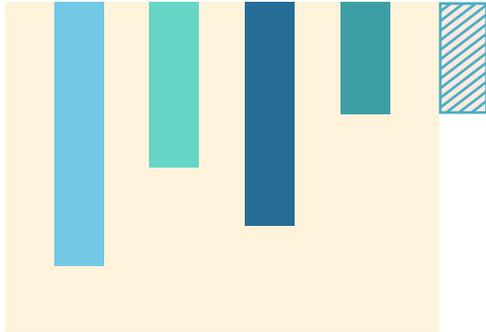
*Scales are functions that map from an **input domain** to an **output range**.* – Mike Bostock

D3 SCALES – PROBLEM

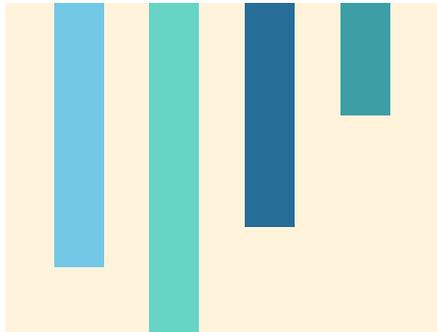
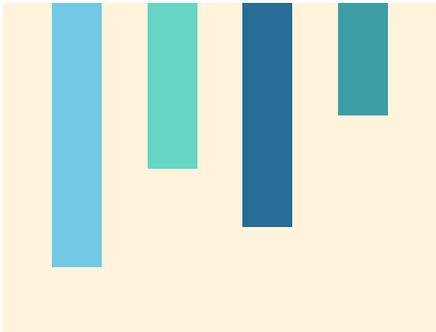
statische Daten



dynamische Daten



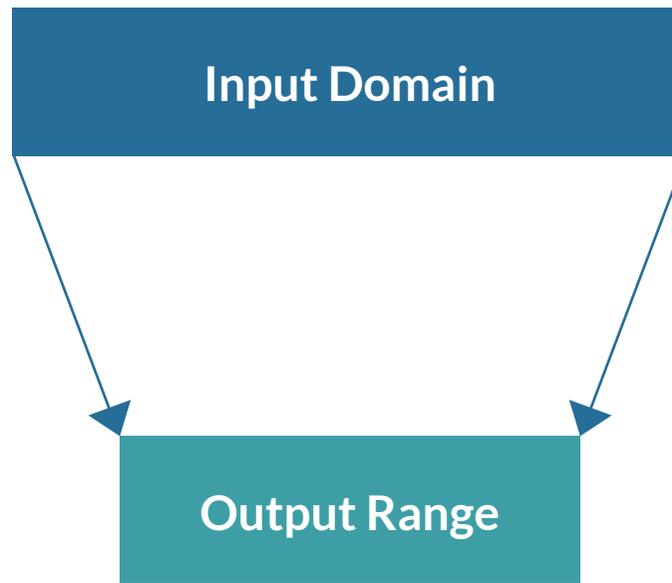
x-Achse



y-Achse



D3 SCALES



D3 SCALES – LINEAR SCALE

Input Domain
z.B. Gebäudehöhe

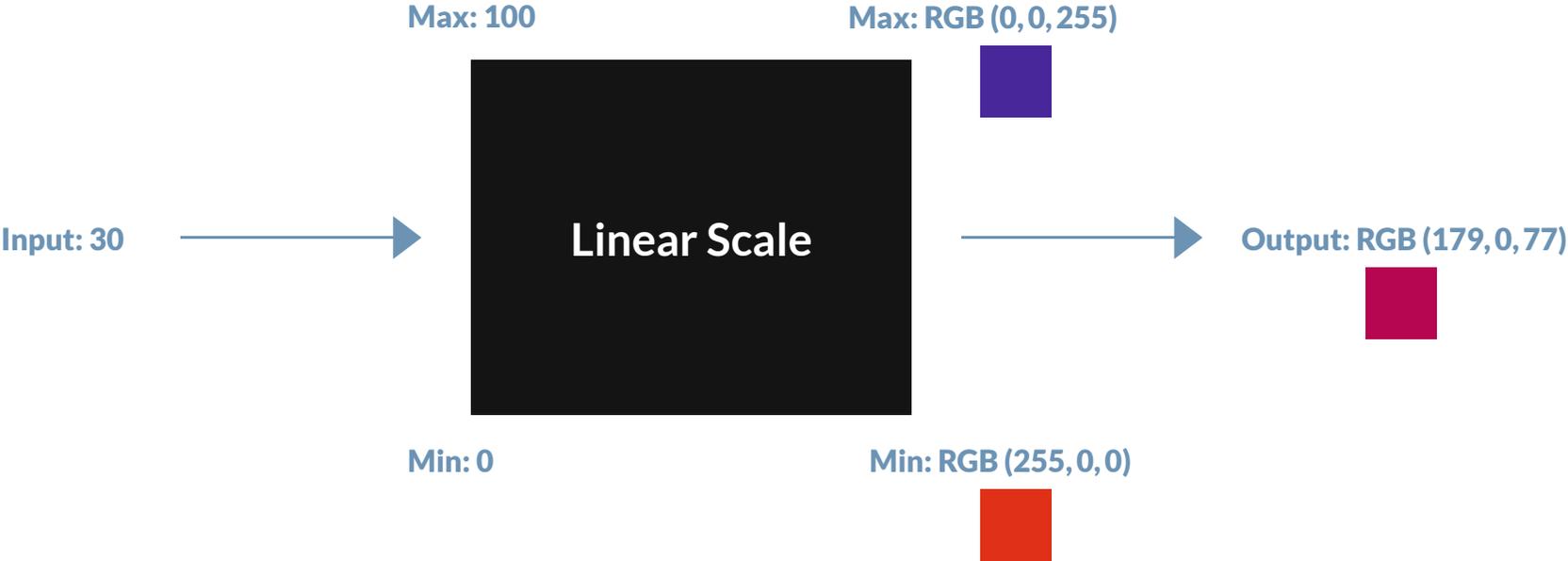
Output Range
z.B. Pixel



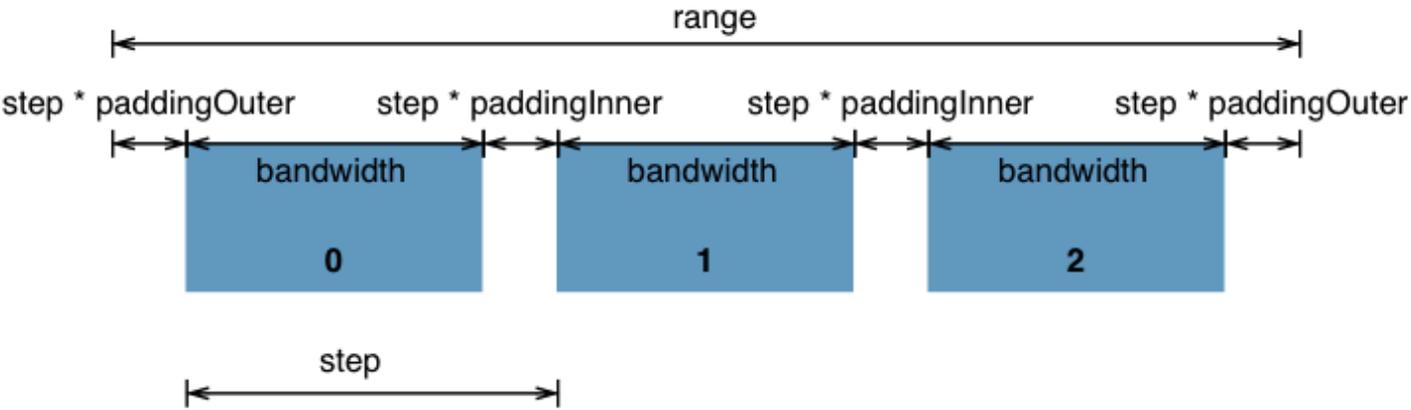
D3 SCALES – LINEAR SCALE

Input Domain
z.B. Prozent

Output Range
z.B. RGB-Werte



D3 SCALES – BAND SCALE



<https://github.com/d3/d3-scale#band-scales>

D3 SCALES – BAND SCALE

Input Domain

z.B. Städte

["Berlin", "Hamburg",
"München", "Köln",
"Bonn", "Essen"]

Inputs

"Hamburg" →
"Essen" →
"Berlin" →
"Duisburg" →



Output Range

z.B. Pixel

PaddingInner: 0.3
PaddingOuter: 0.2

Max: 400

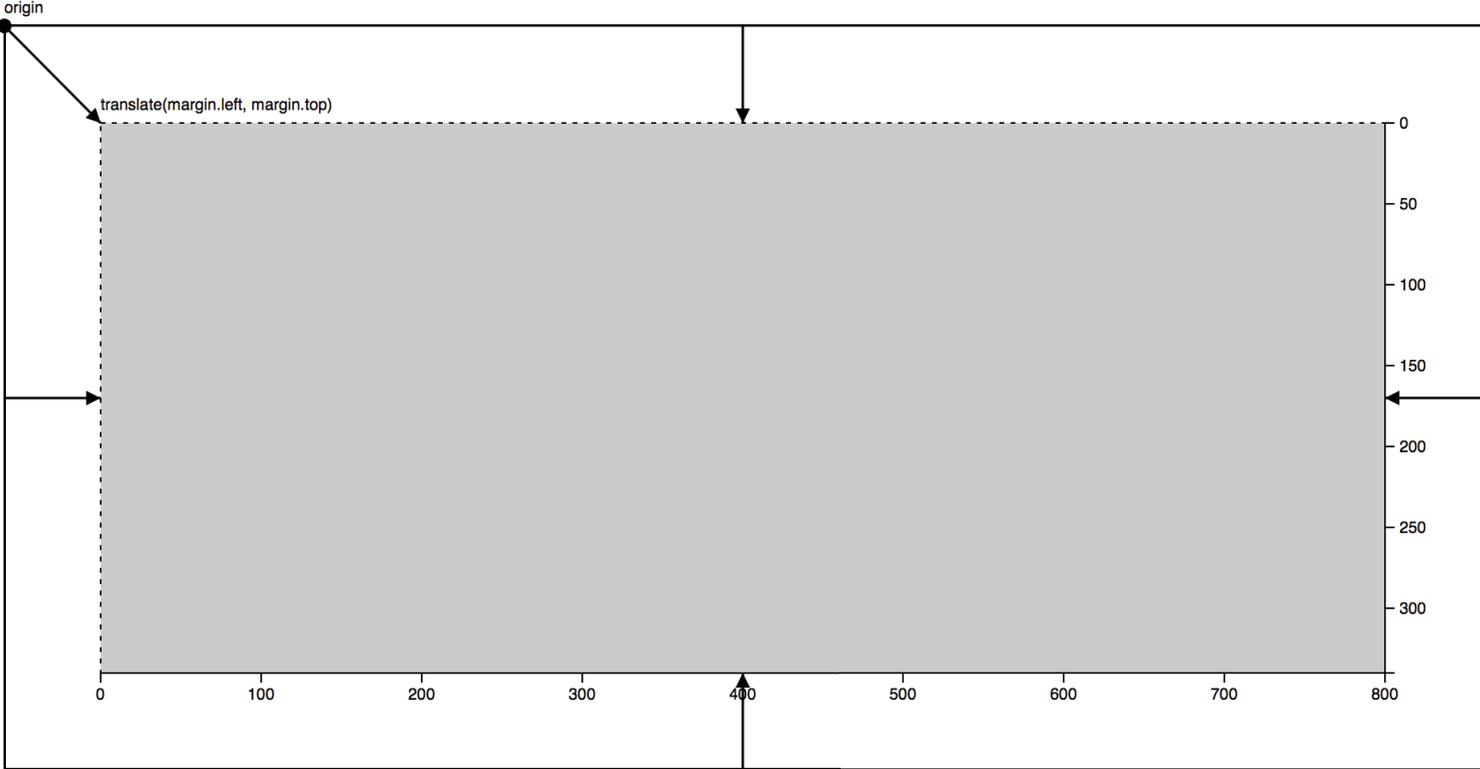
→
→
→
→

Outputs

209
341
13.1
undefined

Min: 0

D3 MARGIN CONVENTION



<https://bl.ocks.org/mbostock/3019563>

DANKE FÜR DIE
AUFMERKSAMKEIT.