

Cheat Sheet D3-Basics

Getting started with D3

D3 steht für *Data-Driven Documents*. Diese Bibliothek ist sehr hilfreich zum Erstellen von Visualisierungen in Javascript, da sie uns viele hilfreiche Funktionen bietet. Im Tutorium haben wir Wetterdaten mit einem simplen Liniendiagramm visualisiert. Die Funktionen, die wir dazu benutzt haben, werden im Folgenden aufgeführt. Dadurch ist es vielleicht ein bisschen einfacher zu verstehen, was passiert, und welche ganz grundlegenden Sachen wir angewandt haben.

HTML-Grundgerüst Die erste Visualisierung, die wir erstellt haben mit D3 benötigt zum Anzeigen eine html-Datei, die wir mit einem beliebigen Browser anzeigen lassen können. Dieses Grundgerüst ist im folgenden dargestellt.

```
1 <!DOCTYPE html>
2 <html lang="de">
3   <head>
4     <meta charset="utf-8" />
5     <title>Seiten-Titel</title>
6     <link rel="optionales_stylesheetsheet" href="./style.css" />
7   </head>
8   <body>
9     <h1>Titel/Text</h1>
10    <div id="Elemente_auf_der_Seite"></div>
11    <script src="https://d3js.org/d3.v6.min.js"></script>
12    <script src="./script.js"></script>
13  </body>
14 </html>
```

Grundlegende D3-Funktionalitäten am Beispiel "simples Liniendiagramm"

In der folgenden Auflistung werden ein paar wichtige Begriffe und Funktionen, die wir kennengelernt haben, vorgestellt. Das sind natürlich nur wenige Funktionen, aber Sie können und sollten für weiteres Arbeiten mit D3 diese und andere Funktionen in der D3-Doku ¹ nachschlagen.

svg steht für **scalable vector graphics** und ist unser Bildelement, in welchem unsere Visualisierung entsteht und abgebildet wird. Dieses Erzeugen wir als variable zu Beginn des Visualisierungsprozesses. D3 ruft dann Funktionen auf, die Objekte erzeugen, manipulieren, usw., welche dann dem svg-Element hinzugefügt werden.

svg.append("g") Da ein svg-Objekt in Javascript nicht direkt transformiert werden kann, kann man das "g" Objekt zum svg-Element hinzufügen und somit dieses somit verändern. Das g steht für group.

d3.select("#Elemente_auf_der_Seite") nach Erzeugung des svg-Elementes sollte dieses in unserem Browser auch erscheinen. Dazu wird dieses von D3 gewählt und anschließend mit dem Aufruf `.append("svg")` hinzugefügt. An der stelle können Größe und andere Eigenschaften mit der `.attr()`-Funktion gesetzt werden,

d3.attr() Setzt Attribute eines ausgewählten Elementes. Attribute können viele verschiedene Dinge sein, die Farbe eines Elementes, die Form, Größe, Position usw.

d3.scaleLinear() Diese Funktion erzeugt uns eine skalierte Linie. Mit einigen anderen Funktionen wie `domain()`, `range()` können wir diese dann anpassen, bevor wir sie mit `append("g")` unserem svg-Element hinzufügen.

d3.domain() Range spezifiziert bestimmte Bereiche, in welchen die Daten liegen, die wir visualisieren wollen.

¹<https://github.com/d3/d3/wiki>

d3.range() setzt dann die visuellen Grenzen der Domain-Daten.

d3.data(), d3.datum() Gegebene Daten werden entweder einem oder mehreren svg-Elementen hinzugefügt. Diese Daten sind die Daten, die wir aus unseren Datensätzen visualisieren wollen

d3.extend() Gibt Minima und Maxima bestimmter Daten-Arrays zurück. Das ist hilfreich, um zum Beispiel den Beginn und das Ende einer Achse zu markieren.

d3.line() Diese Funktion zeichnet unsere Linie. Dazu hängen wir die `.datum()` Funktion an, weil die Linie von den Koordinaten der Daten abhängt, sowie eine Reihe an Attributsfunktionen, um die Eigenschaften der Linie zu setzen.

Hilfreiche Links

- Viele Visualisierungen kann man gut mit einer Vorlage erzeugen, welche man dann beliebig bearbeitet. Viele nützliche Vorlagen findet man hier: <https://www.d3-graph-gallery.com>
- Die offizielle D3-Seite bietet auch Beispiele an, sowie die Doku und API: <https://d3js.org>