

# Funktionsweise der Animation

- Um nun alle Zwischenwerte zu berechnen, müssen wir diese Erhöhung FPS-mal auf den Wert addieren.

Beispiel: FPS = 10 , Erhöhung pro Bild = 40

Wert = 2100 ← **Anfangswert**

Werte für Musterstadt = []

Wiederhole 10 mal: ← **FPS**

Speichere Wert in „Werte für Musterstadt“

Wert = Wert + 40 ← **Änderung pro Bild**

Somit erhalten wir folgende Werte:

2100, 2140, 2180, 2220, 2260, 2300, 2340, 2380, 2420, 2460

| Einwohner | Musterstadt |
|-----------|-------------|
| 1960      | 2100        |
| 1970      | 2500        |
| 1980      | 2800        |

Quelle 2



# Animation in JavaScript

Variablen: FPS = 10, Anfangswert = 2100, Endwert = 2500

Wertänderung pro Bild =  $(\text{Endwert} - \text{Anfangswert}) / \text{FPS}$

Wert = Anfangswert

Werte = []

Wiederhole FPS mal:

Speichere Wert in „Werte“

Wert = Wert + Wertänderung pro Bild

Das gesamte Skript zum Animieren ist zwar knappe 600 Zeilen lang, aber diese Zeilen sind der Kern.

Hier einmal das gesamte Skript um 90 Grad gedreht:

Wertänderung

Endwert

Anfangswert

FPS



```
let ValueDifference = (NextValue - CurrentValue) / FramesPerValue
ccc = 0
while (ccc < FramesPerValue){
  DataObject.values.push(CurrentValue + ValueDifference * ccc)
  DataObject.rowNames.push(csvMatrix[cc+1][0])
  ccc += 1
}
```

Speichern des Werts

Quelle 1

CSV in Listen umwandeln

Grafikfunktionen

Liniendiagramme animieren

Ein-Balken-Diagramme animieren

Finden des Trenners

Listen in Objekte umwandeln

Balkendiagramme animieren

Flächendiagramme animieren