

Moderne Webentwicklung

Andreas Roth

quēo

Eine Reise durch die Zeit

In 3 Akten



```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="UTF-8" />
    <meta
      name="viewport"
      content="width=device-width, maximum-scale=1.0, minimum-scale=1.0, initial-scale=1.0"
    />
    <title>Wikipedia, die freie Enzyklop&auml;die</title>
    <link rel="apple-touch-icon" href="/img/wikipedia.png" />
    <link rel="stylesheet" media="screen" type="text/css" href="style.css" />
    <script type="text/javascript" src="js/jquery-3.4.1.min.js"></script>
    <script type="text/javascript" src="js/jquery.cookie.min.js"></script>
    <script type="text/javascript" src="suggest.js"></script>
  </head>

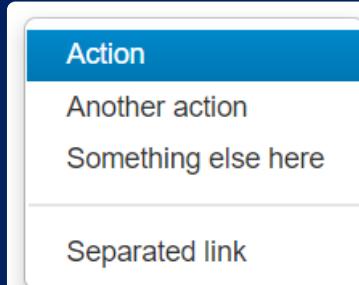
  <body>
    <div id="WMDE-Banner-Container"></div>
    <div id="main">
      <div id="mainbox">
        <div class="wikipedia-logo">
          <a href="https://de.wikipedia.org/">
<html lang="de">
  <head>
    <meta charset="UTF-8" />
    <meta
      name="viewport"
      content="width=device-width, maximum-scale=1.0, minimum-scale=1.0, initial-scale=1.0"
    />
    <title>Wikipedia, die freie Enzyklop&auml;die</title>
    <link rel="apple-touch-icon" href="/img/wikipedia.png" />
    <link rel="stylesheet" media="screen" type="text/css" href="style.css" />
    <script type="text/javascript" src="js/jquery-3.4.1.min.js"></script>
    <script type="text/javascript" src="js/jquery.cookie.min.js"></script>
    <script type="text/javascript" src="suggest.js"></script>
  </head>

  <body>
    <div id="WMDE-Banner-Container"></div>
    <div id="main">
      <div id="mainbox">
        <div class="wikipedia-logo">
          <a href="https://de.wikipedia.org/">
<html lang="de">
  <head>
    <meta charset="UTF-8" />
    <meta
      name="viewport"
      content="width=device-width, maximum-scale=1.0, minimum-scale=1.0, initial-scale=1.0"
    />
    <title>Wikipedia, die freie Enzyklop&auml;die</title>
    <link rel="apple-touch-icon" href="/img/wikipedia.png" />
    <link rel="stylesheet" media="screen" type="text/css" href="style.css" />
    <script type="text/javascript" src="js/jquery-3.4.1.min.js"></script>
    <script type="text/javascript" src="js/jquery.cookie.min.js"></script>
    <script type="text/javascript" src="suggest.js"></script>
  </head>

  <body>
    <div id="WMDE-Banner-Container"></div>
    <div id="main">
      <div id="mainbox">
        <div class="wikipedia-logo">
          <a href="https://de.wikipedia.org/">

<ul class="dropdown-menu" role="menu">
  <li><a href="#">Action</a></li>
  <li><a href="#">Another action</a></li>
  <li><a href="#">Something else here</a></li>
  <li class="divider"></li>
  <li><a href="#">Separated link</a></li>
</ul>

<script
src="https://code.jquery.com/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
```



Bootstrap & Co

UI-Bibliotheken heben die Qualität

- Wiederverwendbare Komponenten
- Anpassbarkeit durch LESS bzw. SASS
- Mobile-First wird zum Trend
 - Seit 2017 mobile Traffic über 50%
 - Google indiziert nur noch über die mobile Ansicht
- Websites müssen mobile & touch-fähig sein
 - Kein Hover
 - Kein Rechtsklick

Utility-First CSS bzw. Atomic CSS

Du musst es probieren

- Kein Separation of Technology mehr
- CSS wird wiederverwendbar
- Colocation



"Tailwind CSS is the only framework that I've seen scale on large teams. It's easy to customize, adapts to any design, and the build size is tiny."

Sarah Dayan
Staff Engineer, Algolia

```
ooo
1 <figure class="bg-gray-100 rounded-xl">
2   
4     <blockquote>
5       <p class="text-lg">
6         "Tailwind CSS is the only framework that I've seen scale
7           on large teams. It's easy to customize, adapts to any design,
8             and the build size is tiny."
9       </p>
10      </blockquote>
11      <figcaption>
12        <div>
13          Sarah Dayan
14        </div>
15        <div>
16          Staff Engineer, Algolia
17        </div>
18      </figcaption>
19    </div>
20  </figure>
21
```

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="UTF-8" />
    <meta
      name="viewport"
      content="width=device-width, maximum-scale=1.0, minimum-scale=1.0, initial-scale=1.0"
    />
    <title>Wikipedia, die freie Enzyklop&auml;die</title>
    <link rel="apple-touch-icon" href="/img/wikipedia.png" />
    <link rel="stylesheet" media="screen" type="text/css" href="style.css" />
    <script type="text/javascript" src="js/jquery-3.4.1.min.js"></script>
    <script type="text/javascript" src="js/jquery.cookie.min.js"></script>
    <script type="text/javascript" src="suggest.js"></script>
  </head>

  <body>
    <div id="WMDE-Banner-Container"></div>
    <div id="main">
      <div id="mainbox">
        <div class="wikipedia-logo">
          <a href="https://de.wikipedia.org/"></a>
        </div>
        <div id="maincontent">
          <div id="form">
```

Stufe 1

- Semantische Klassen
- Wenige, riesige (S)CSS-Files
- Anpassung häufig in HTML & CSS nötig

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'app-product-alerts',
  templateUrl: './product-alerts.component.html',
  styleUrls: ['./product-alerts.component.css']
})
export class ProductAlertsComponent implements OnInit {
  constructor() { }

  ngOnInit() {
  }
}
```

Stufe 2

- Denken in Komponenten
- Trennung von HTML, CSS und JS

```
<template>
  <p>{{ greeting }} World!</p>
</template>
```

```
<script>
module.exports = {
  data: function() {
    return {
      greeting: "Hello"
    };
  }
};
</script>
```

```
<style scoped>
p {
  font-size: 2em;
  text-align: center;
}
</style>
```

Stufe 3

- Reduktion der Distanz zwischen HTML, CSS & JS

```
export function ContactElement() {  
  const greeting = "Hello";  
  return (  
    <p class="text-2xl text-center">  
      {greeting} World!  
    </p>  
  );  
}
```

Stufe 4

- Verschmelzung von HTML, CSS und JS

A photograph of a modern architectural structure, likely a residential building, featuring a complex, angular facade made of light-colored panels. The building has many windows of various sizes and orientations. The perspective is from a low angle looking up at the building against a clear blue sky.

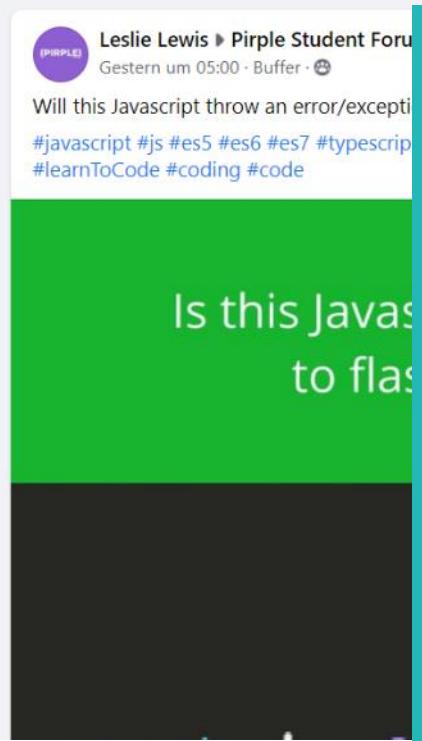
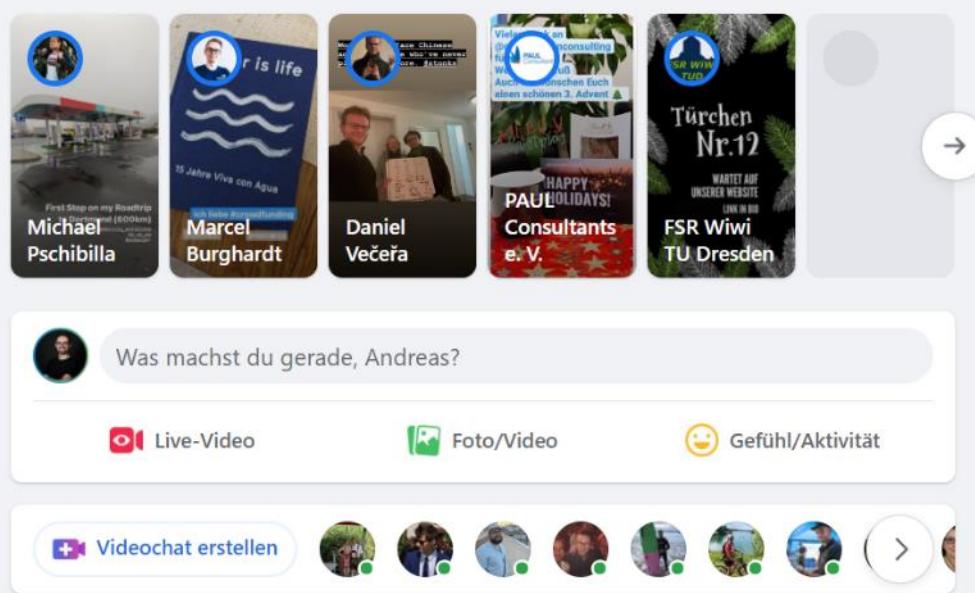
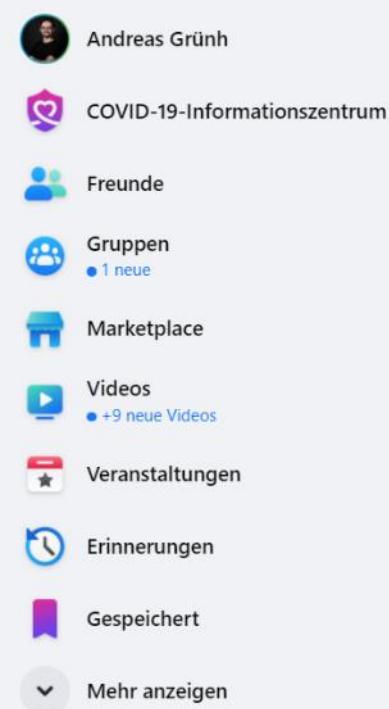
Neue Features

- Flex & Grid
- Custom Properties
- Aspect-Ratio



HTML Browser als Rendering Engine

Von statischen Dokumenten zu dynamischen Applikationen



Gesponsert

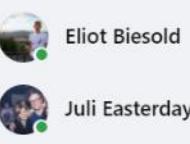


Exclusive Fabric Networking For Dummies Guide



**Exclusive Fabric
Networking For Dumb
Guide**
learn.extremenetworks.com

Kontakte



Blogs, Gästebücher & Shops

- Web 2.0
 - Interaktivität wird Teil des Webs
 - Native APIs: GPS, Kamera, Mikrofon

E-commerce Reporting Template

Productivity Tools

- Ohne Installation
- Online Collaboration
- Realtime

Codesandbox interface showing a React application for a "Dataviz with drilldown by React & D3".

The code in `index.js` uses React, useState, useEffect, ReactDOM, d3, styled-components, faker, and randomNormal to generate hierarchical data and render it as a pie chart with mouseover effects.

```
import React, { useState, useEffect } from "react";
import ReactDOM from "react-dom";
import "./styles.css";
import * as d3 from "d3";
import faker from "faker";
import styled from "styled-components";

function generateData(level) {
  const Num = d3.randomUniform(1, 10)(); //randomUniform is a function of D
  return d3.range(Num).map(i => ({
    value: Math.abs(d3.randomNormal()()),
    id: `${level}-${i}`,
    level: level,
    index: i,
    name: faker.company.companyName(),
    children: level > 0 ? generateData(level - 1) : []
  }));
}

const Path = styled.path`
  fill: ${props => d3.schemePastel2[props.index]};
  cursor: pointer;
`;

const Arc = ({ arcData }) => {
  const [addedRadius, setAddedRadius] = useState(0);
  const arc = d3
    .arc()
    .innerRadius(25 + addedRadius / 3)
    .outerRadius(125 + addedRadius);

  function mouseOver() {
    setAddedRadius(20);
  }

  function mouseOut() {
    setAddedRadius(0);
  }

  return (
    <Path
      d={arc(arcData)}
      index={arcData.data.index}
      onMouseOver={mouseOver}
      onMouseOut={mouseOut}
    />
  );
};

function App() {
  const [data, setData] = useState(generateData(3));
  const [radius, setRadius] = useState(0);

  useEffect(() => {
    const chart = d3.select("svg");
    const chartRadius = chart.attr("r");
    const chartCenter = chart.attr("cx") + chartRadius;
    const chartOuterRadius = chartRadius + radius;

    const arc = d3
      .arc()
      .innerRadius(chartRadius)
      .outerRadius(chartOuterRadius);

    const path = chart.selectAll("path").data(data);
    path
      .attr("fill", "white")
      .attr("stroke", "#ccc")
      .attr("strokeWidth", 2);
    path = path.enter().append("path");
    path
      .attr("d", arc)
      .attr("fill", "white")
      .attr("stroke", "#ccc")
      .attr("strokeWidth", 2);
  }, [data, radius]);
}

ReactDOM.render(<App/>, document.getElementById("root"));
```

The browser preview shows a pie chart with three segments: light blue, orange, and green. A mouse cursor is hovering over the green segment, which has a white center and a thick black border.

Cloud First

- Performance
- File System API

☰

 
Hinzufü... wesbos

 **Stadt Dresden** @stadt_dresden · 1 Std.
Corona: Ab Montag, 14. Dezember gelten
landesweit einheitliche Regelungen dlvr.it/RnYM6Z


Q 3 T 1 H 6

Alex retweetete

 **Der Gazetteur** @dergazett... · 2 Std.
Einzelhandel am Montag und Dienstag.
#Lockdown


Q 23 T 321 H 2.012

☰

Neueste Tweets

 **Stadt Dresden** @stadt_... · 1 Std. ...
Corona: Ab Montag, 14. Dezember gelten
landesweit einheitliche Regelungen dlvr.it/RnYM6Z


Q 3 T 1 H 6

Alex hat retweetet

 **Der Gazetteur** @dergazett... · 2 Std. ...
Einzelhandel am Montag und Dienstag.
#Lockdown


Q 23 T 335 H 2.068

Adam Wathan hat retweetet

 **Henrik Larsson** @Henkisch · 4 Std. ...
I think I've fallen in love with @tai...
indefinitely! I've written a little summary of



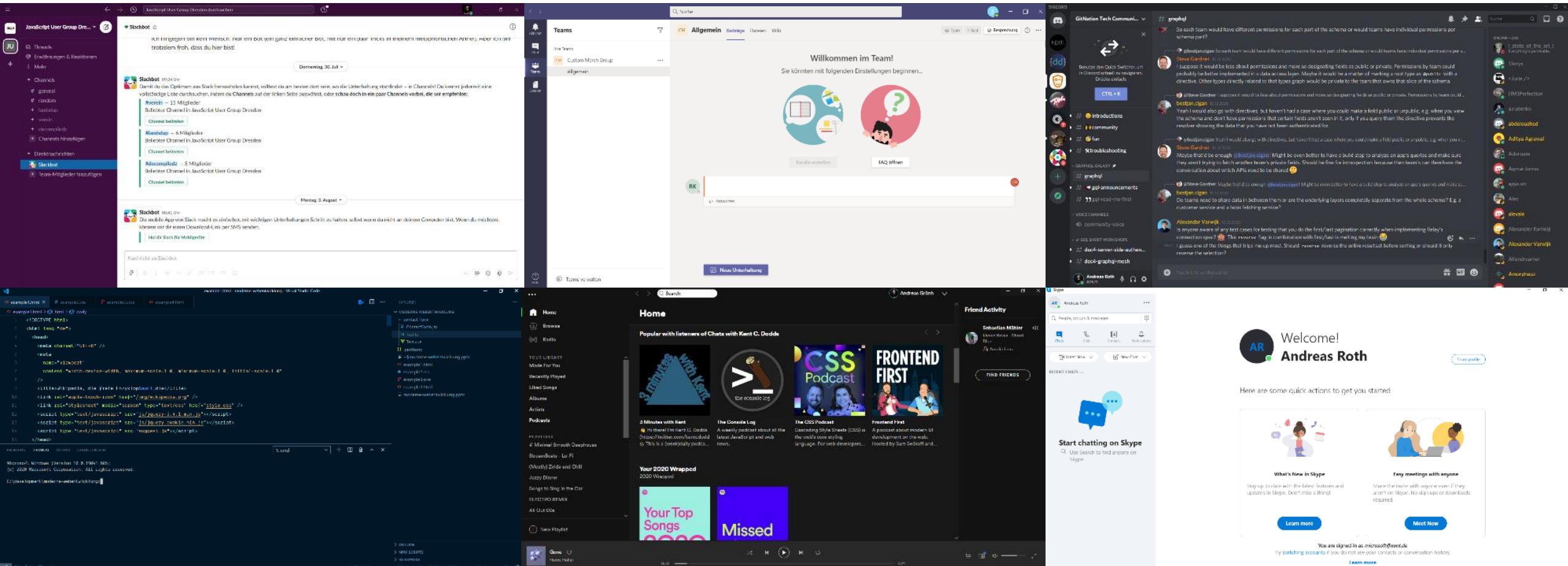
   

PWA

Die Grenze zwischen Web und App verschwimmt

quo

HTML verlässt den Browser - oder?



queo

A close-up photograph of a young lion cub's head, positioned in the center of a white rectangular overlay. The cub is captured mid-yawn, with its mouth wide open showing its tongue and teeth. Its eyes are closed, and its ears are perked up. The background consists of tall, green, out-of-focus grass.

JavaScript Eine Sprache wird erwachsen

Die Geschichte der meistverwendeten Programmiersprache der Welt

```
console.log([] + []);
//> ""

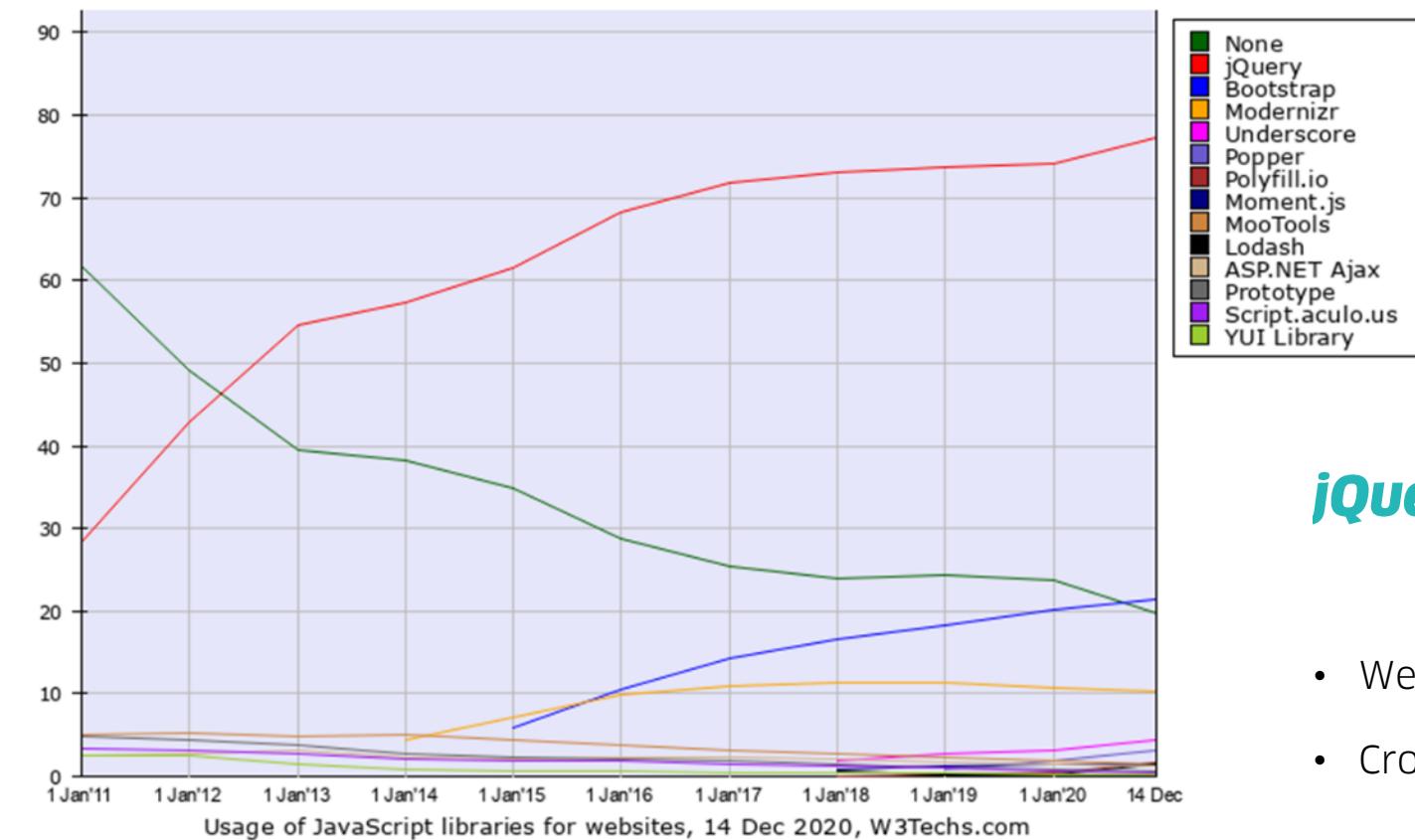
console.log([] + {});
//> "[object Object]"

console.log(11 + "12");
//> "1112"

console.log(NaN === NaN);
//> false
```

In 10 Tagen zur vollwertigen Programmiersprache?

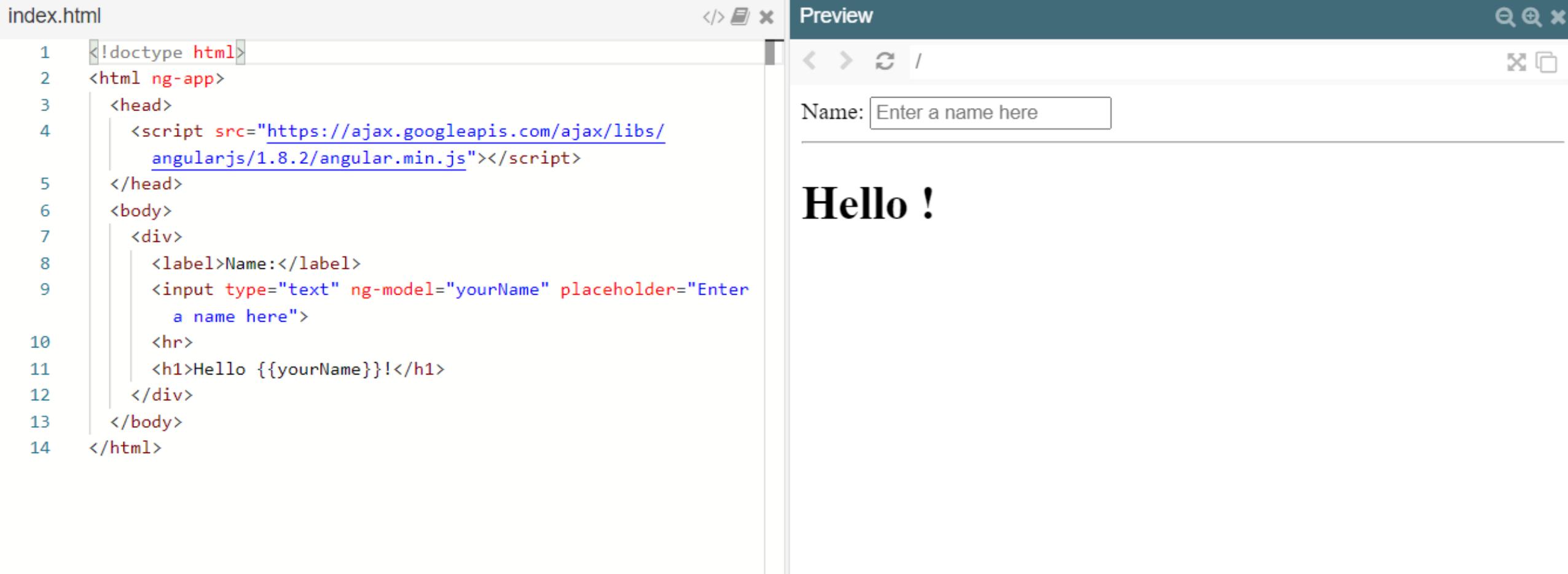
- Für einfachste Interaktionen gemacht
 - Dynamisch und schwach typisiert
 - Möglichst Fehlertolerant
- Viele Fallstricke



jQuery im Kampf gegen Browser

- Weniger Code
- Cross-Browser Kompatibilität

Angular.js als Vorreiter für SPAs

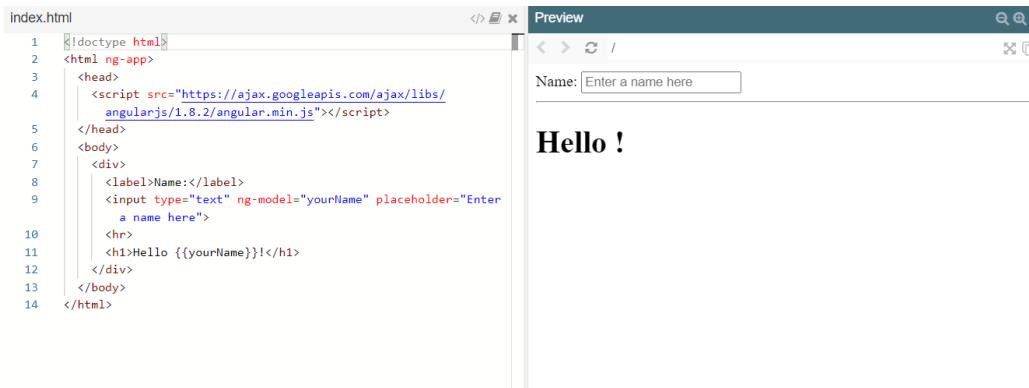


The screenshot shows a code editor window with a preview tab. The code editor displays the file `index.html` containing the following Angular.js application:

```
1 <!doctype html>
2 <html ng-app>
3   <head>
4     <script src="https://ajax.googleapis.com/ajax/libs/
      angularjs/1.8.2/angular.min.js"></script>
5   </head>
6   <body>
7     <div>
8       <label>Name:</label>
9       <input type="text" ng-model="yourName" placeholder="Enter
      a name here">
10      <hr>
11      <h1>Hello {{yourName}}!</h1>
12    </div>
13  </body>
14 </html>
```

The preview tab shows a browser window with the application running. It features a text input field labeled "Name:" with the placeholder "Enter a name here". Below the input field, the text "Hello !" is displayed.

Angular.js als Vorreiter für SPAs



The screenshot shows a code editor with a file named 'index.html' and a browser preview window. The code in 'index.html' is:

```
index.html
1 <!doctype html>
2 <html ng-app>
3   <head>
4     <script src="https://ajax.googleapis.com/ajax/libs/
      angularjs/1.8.2/angular.min.js"></script>
5   </head>
6   <body>
7     <div>
8       <label>Name:</label>
9       <input type="text" ng-model="yourName" placeholder="Enter
      a name here">
10      <hr>
11      <h1>Hello {{yourName}}!</h1>
12    </div>
13  </body>
14 </html>
```

The browser preview shows a simple form with a label 'Name:' and an input field placeholder 'Enter a name here'. Below the form, the text 'Hello !' is displayed.

Design Patterns im Frontend

- MVVM
- Dependency Injection
- Modul-System
- Event Bus

Package Manager

Die schnellste Entwicklung ist keine Entwicklung



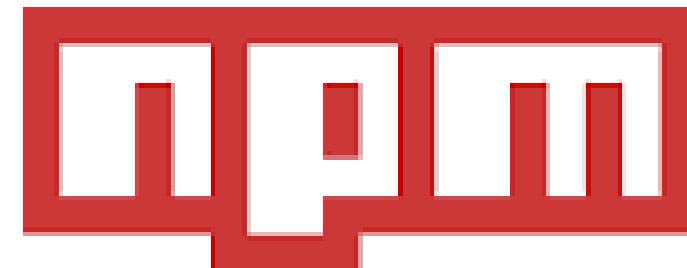
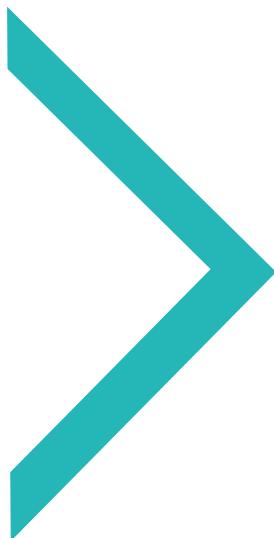
Bower für jQuery, Bootstrap & Co

Package Manager

Die schnellste Entwicklung ist keine Entwicklung



Bower für jQuery, Bootstrap & Co

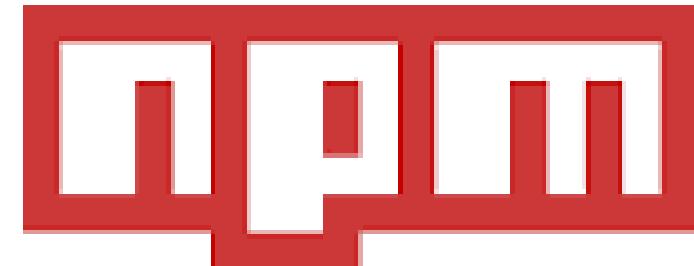


npm als Sieger

Package Manager

Die schnellste Entwicklung ist keine Entwicklung

Über 1.000.000 Pakete

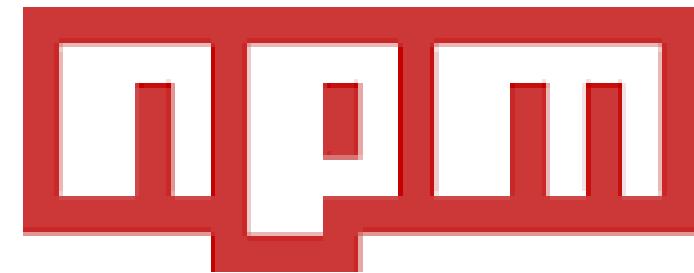


npm als Sieger

Package Manager

Die schnellste Entwicklung ist keine Entwicklung

**Knapp 50.000.000.000
monatliche Downloads**



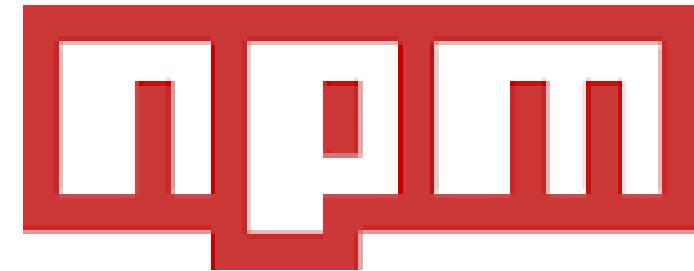
npm als Sieger

Package Manager

Die schnellste Entwicklung ist keine Entwicklung

Wild Wild West

→ **LeftPad Fiasko**



npm als Sieger

Build Chains

Was geschrieben wird kommt lange nicht mehr im Browser an



Gulp & Grunt

- Manuelle Auflistung von files
- Einbindung von zusätzlichen Tools

6 to 5, SASS, Autoprefixer & Uglify

Build Chains

Was geschrieben wird kommt lange nicht mehr im Browser an



Gulp & Grunt

Manuelle Auflistung von files
Einbindung von zusätzlichen Tools

6 to 5, SASS, Autoprefixer & Uglify



Webpack, Parcel & Rollup

Analyse & Optimierung
Einbindung von Compilern wie **Babel**
oder **TypeScript**
Entwicklungsserver mit HMR

ES2015 und TC39

Idee

Erste Diskussionen zu neuen Sprachfeatures

Vorschlag

Vorteile, Lösungsansätze & Herausforderungen

Entwurf

Syntax & Semantik

Kandidat

Feedback durch Implementierungen und Nutzer

Standard

Teil der Spezifikation

```
type Type = "cat" | "dog";

function makeNoise(animalType: Type) {
  const noise =
    animalType === "cat" ? "Meow" : "Woof";
  const message = `The animal said: ${noise}`;

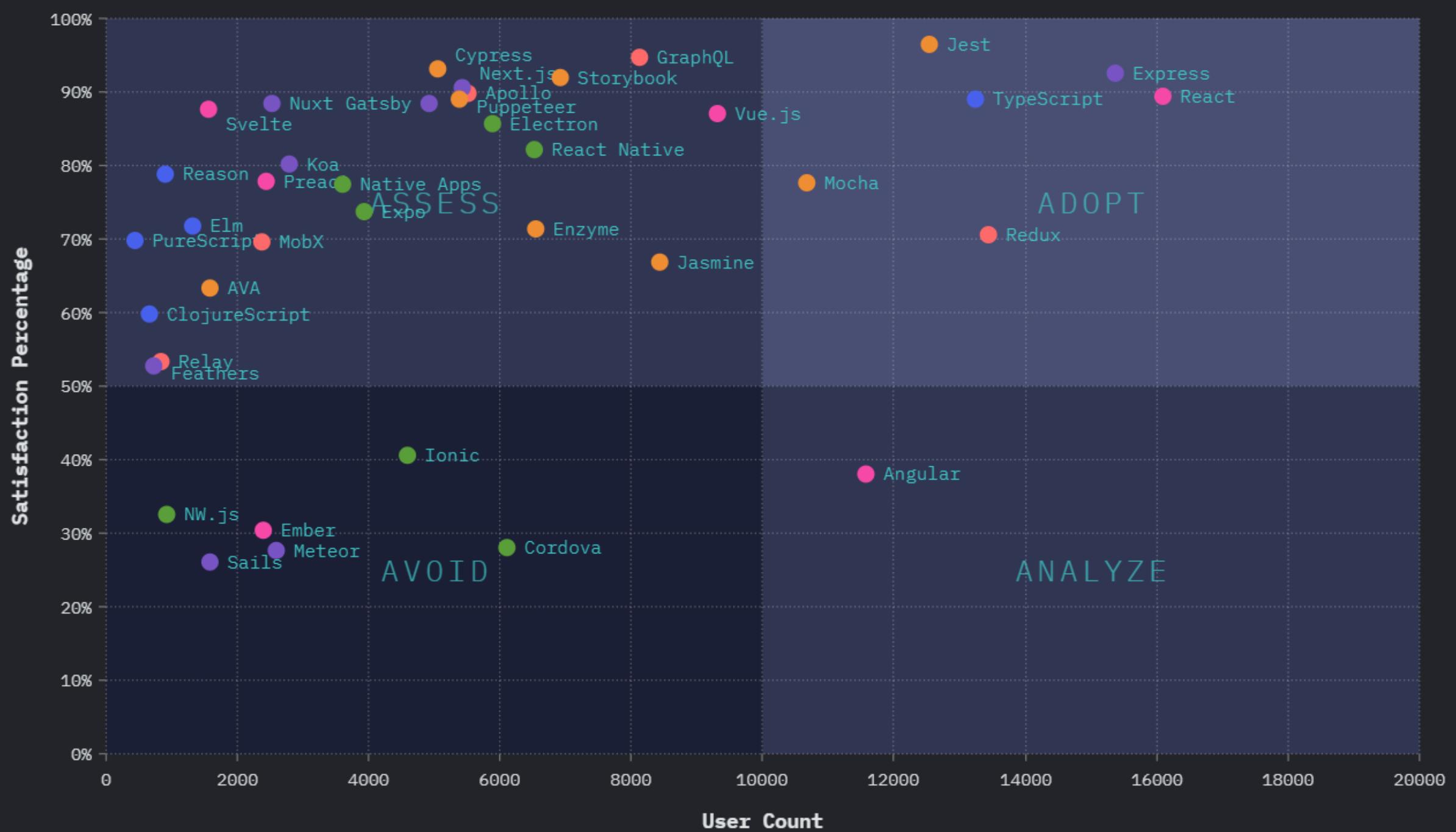
  return message;
}

makeNoise("cat").repeat(5);
```

TypeScript

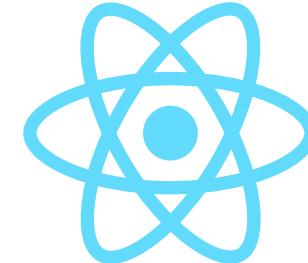
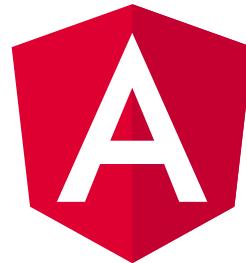
Statische Typen für komplexe Projekte

- Drastische Reduktion von Laufzeit Fehlern
- Extrem verbesserte Developer Experience
- Pragmatische Ansätze durch loose-Mode und any möglich
- Funktionale und Objektorientierte Ansätze





Die großen 3



Größe

Ab ca. 170kb

API-Oberfläche

Groß

Ab ca. 130kb
(ab 20kb mit Preact)

Ab ca. 80kb

Ökosystem

Mittel

Klein

Mittel

TypeScript

Gut (nativ)

Sehr gut
(community)

Gut (nativ)

Guidelines

Stark

Schwach

Mittel

JavaScript verlässt den Browser

Electron
Node.js

React Native

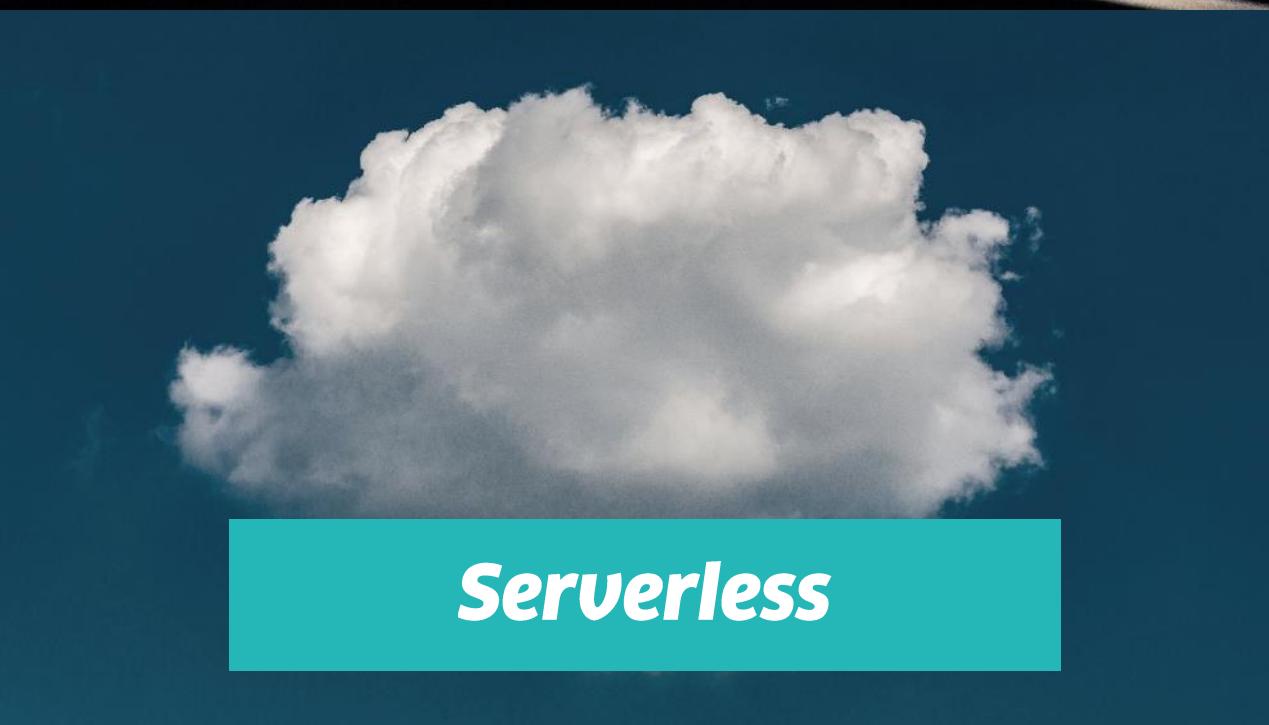




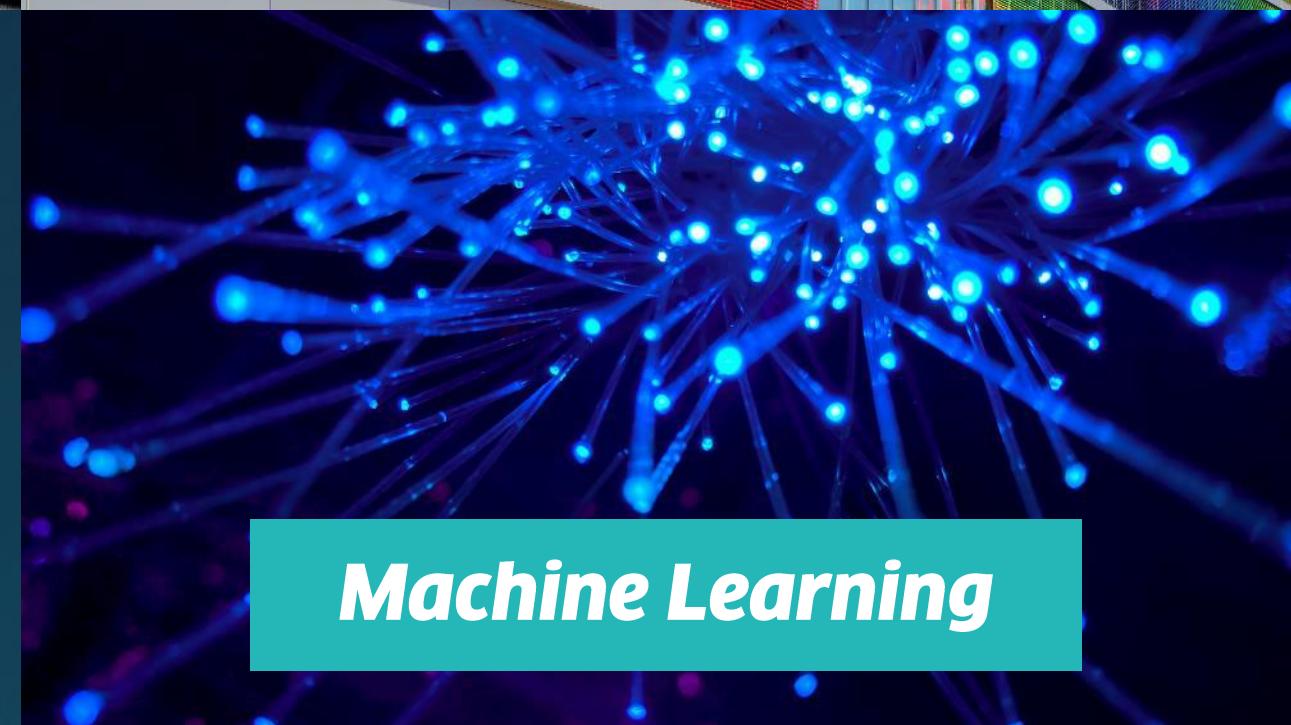
WASM & WebGL



Accessibility

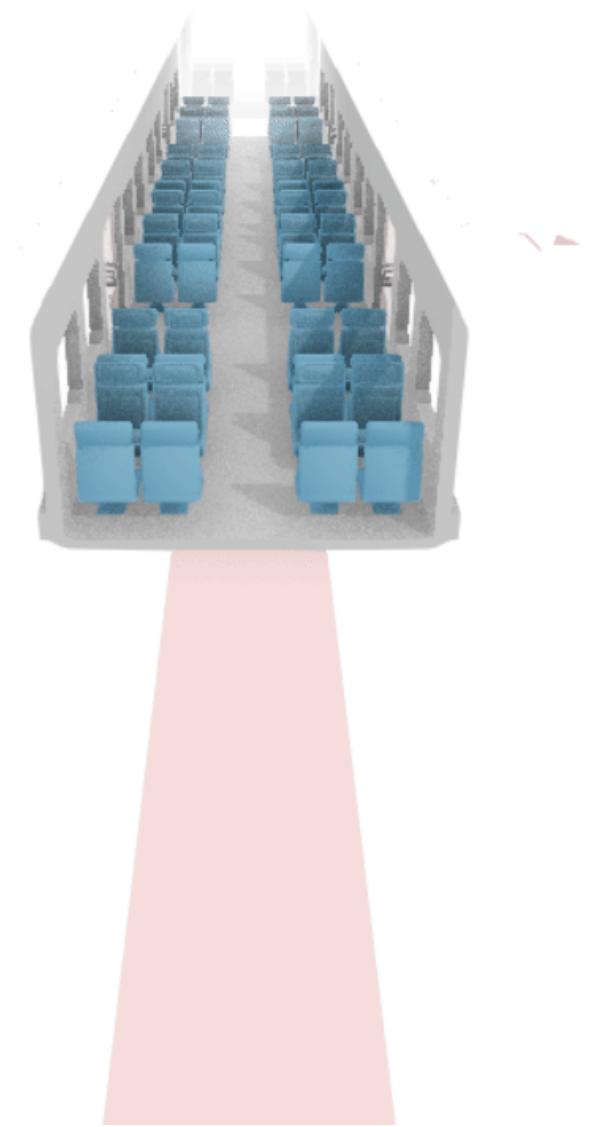


Serverless



Machine Learning

```
48      <Quarter color={color} position={[1.2, -0.45, 9.75]} />
49    </group>
50  )
51
52 const Cabin = ({ color = 'white', seatColor = 'white', name, ...
53   const [geometries, center] = useModel('/cabin.glb')
54   return (
55     <group {...props}>
56       <Text centerX={false} color="lightcoral" size={0.6} posit
57         {name}
58       </Text>
59       <group position={center}>
60         {geometries.map((geom, index) => (
61           <mesh key={geom.uuid} geometry={geom} castShadow rece
62             <meshPhysicalMaterial {...material} color={index ==
63               </mesh>
64             ))}
65         </group>
66         <Row color={seatColor} />
67         <Row color={seatColor} position={[0, 0, -1.9]} />
68         <Row color={seatColor} position={[0, 0, -6.6]} />
69         <Row color={seatColor} position={[0, 0, -8.5]} />
70         <Row color={seatColor} position={[0, 0, -11]} />
71         <Row color={seatColor} position={[0, 0, -12.9]} />
72         <Row color={seatColor} position={[0, 0, -17.6]} />
73         <Row color={seatColor} position={[0, 0, -19.5]} />
74       </group>
```

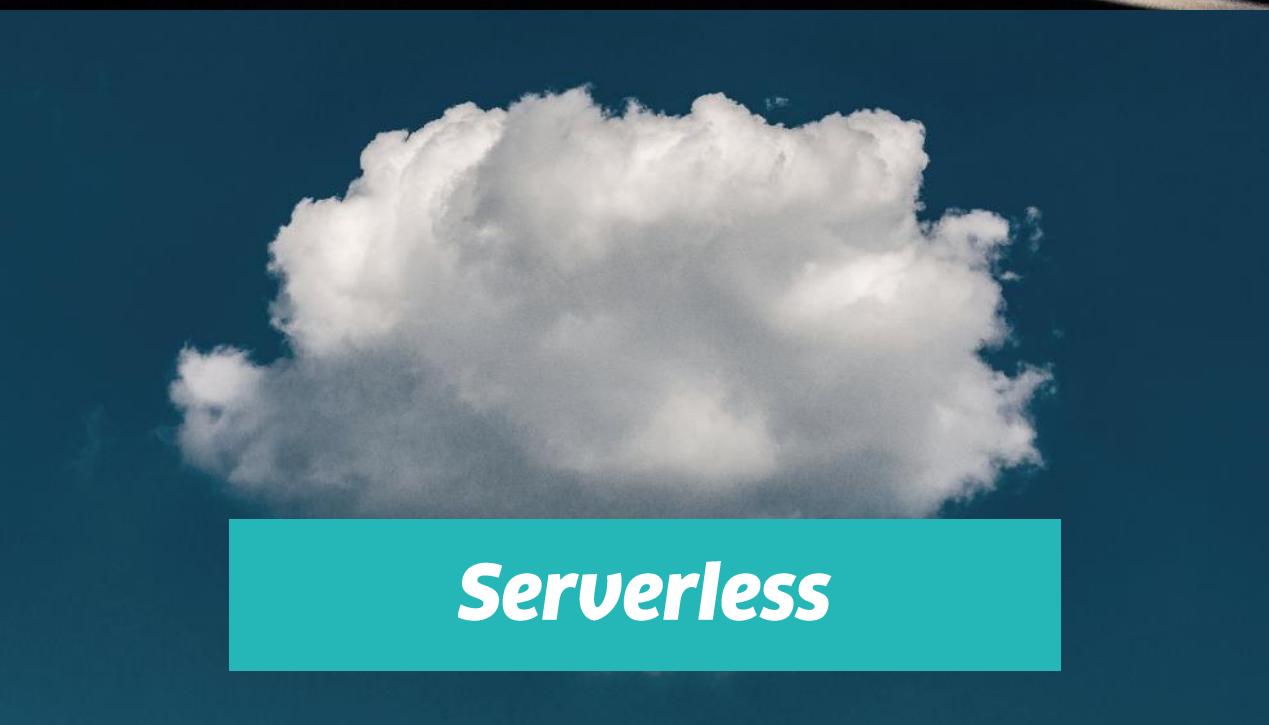




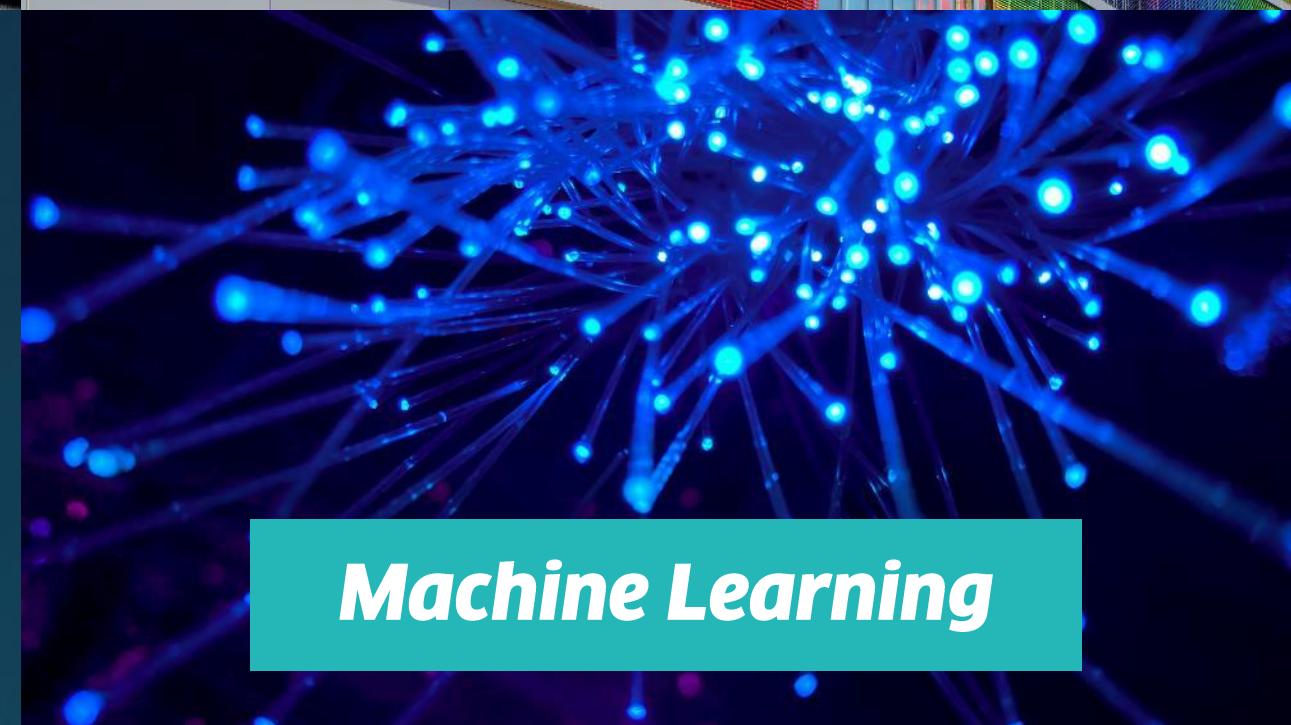
WASM & WebGL



Accessibility



Serverless



Machine Learning

The background of the image shows a modern building's exterior wall. The wall is covered in numerous vertical panels, each with a different color, creating a vibrant, multi-colored pattern. The colors transition through various hues, including red, orange, yellow, green, blue, and purple. The panels are arranged in a grid-like fashion, with some panels being taller than others, creating a textured, layered effect. The building appears to be a large, possibly residential or office complex, with the colorful facade being a prominent feature.

Accessibility

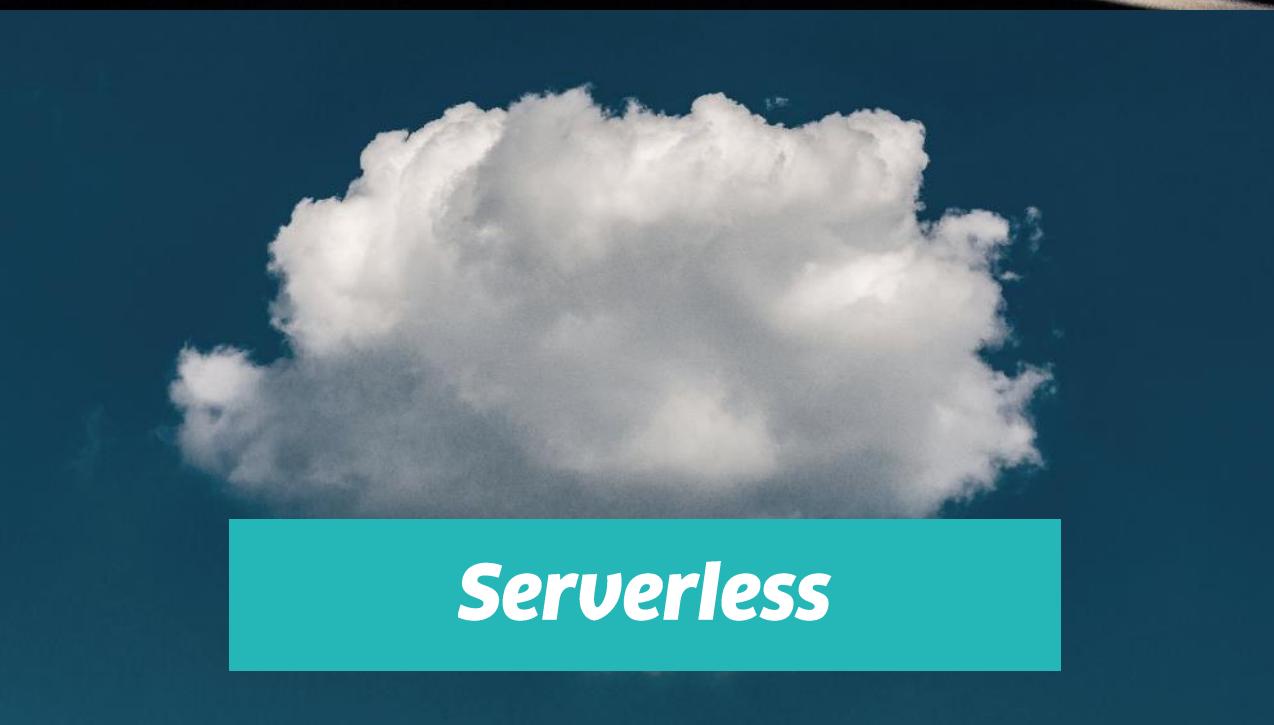
20% sind betroffen



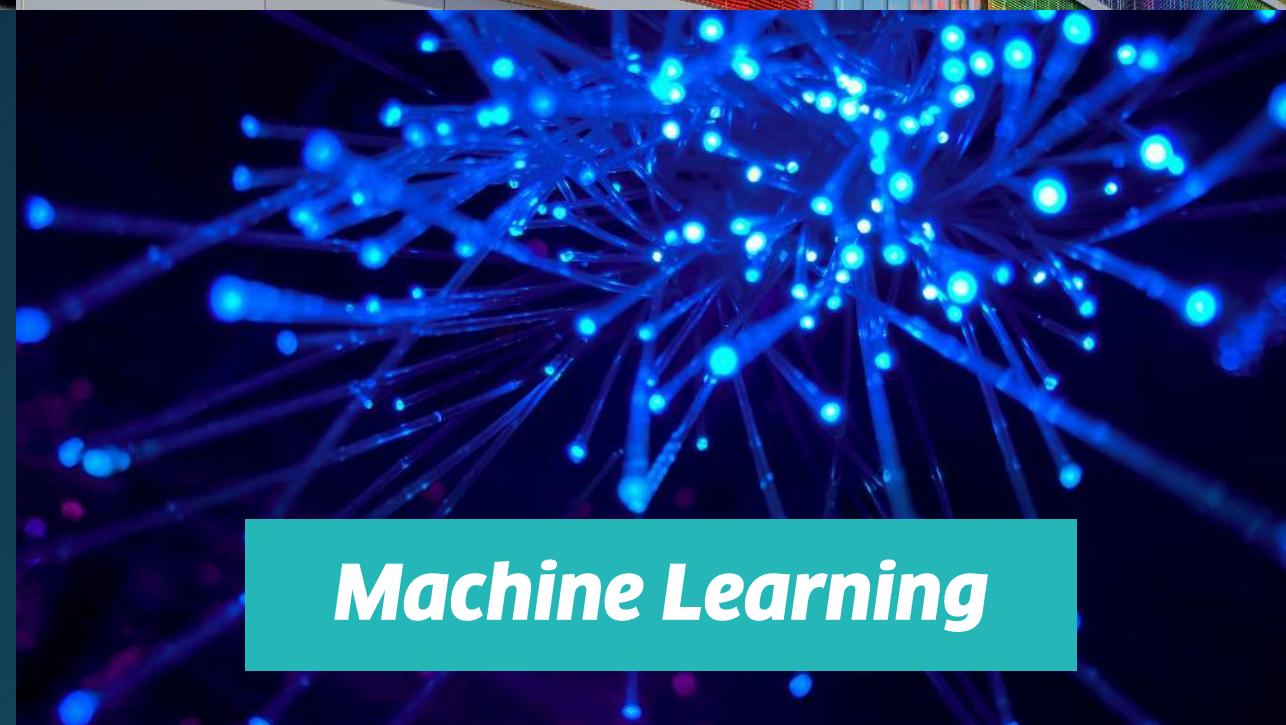
WASM & WebGL



Accessibility



Serverless



Machine Learning

Gemeinsam orientieren, starten & anwenden

**Andreas Roth
Trainer / Consultant**

queo GmbH
+49 163 492 243 5
a.roth@queo-group.com
www.queo.de

