FRUSTIERENDE WEB-ENTWICKLUNG

mit Fallbeispielen in React :)

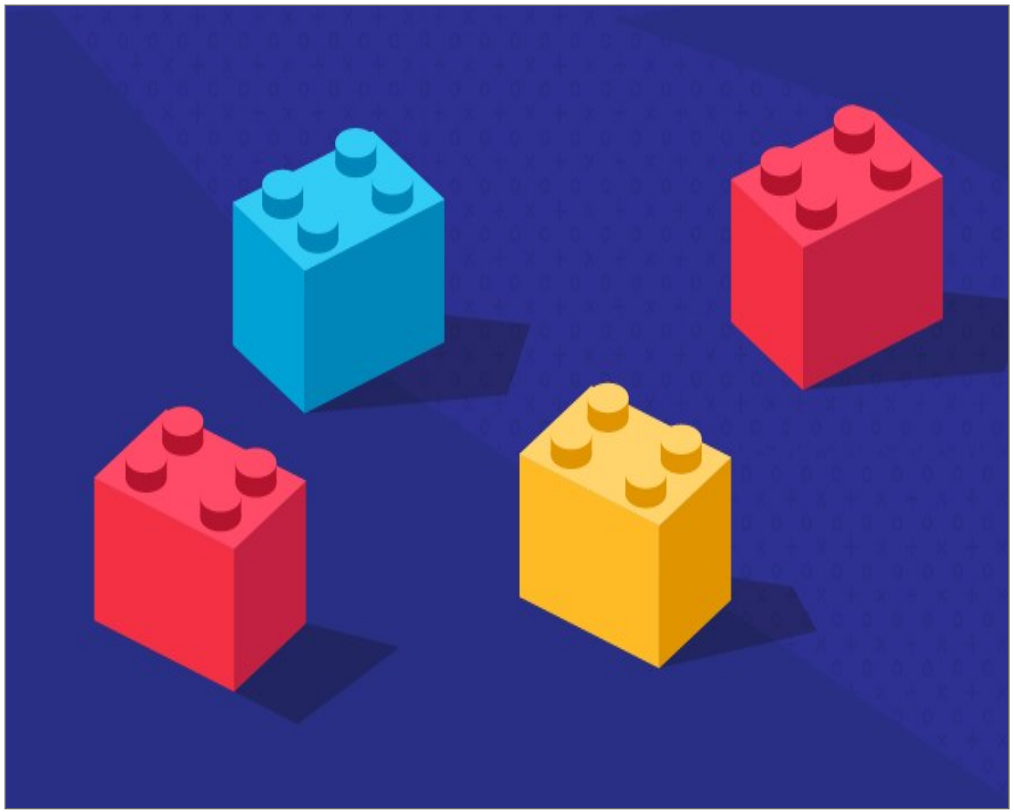
ZU MIR

- Artur Zimmermann
- Master in Informatik (Juli 2019)
- Vollzeit bei Ambient (Mai 2019)
- Diverse Werkstudentenstellen (Galaria Kaufhof, Questback, SWYP GmbH, etc.)

MOTIVATION FÜR VORTRAG

- Die selbe Melody, nur anderer Text
- "Wir wollen von React weg"
- Der Fehler ist nicht React exklusiv
- Alle Frameworks, Library, etc. sind Tools
- Anhand von React als Beispiel

MIT REACT ENTWICKELT MAN WEB-KOMPONENTEN



https://www.sitepen.com/blog/web-components-in-2018/

und das alles schon seit 2013

WEB-COMPONENTS

NATIVE WEB-KOMPONENTEN BESTEHEN AUS:

- HTML Templates
- Custom Elements
- Shadow DOM

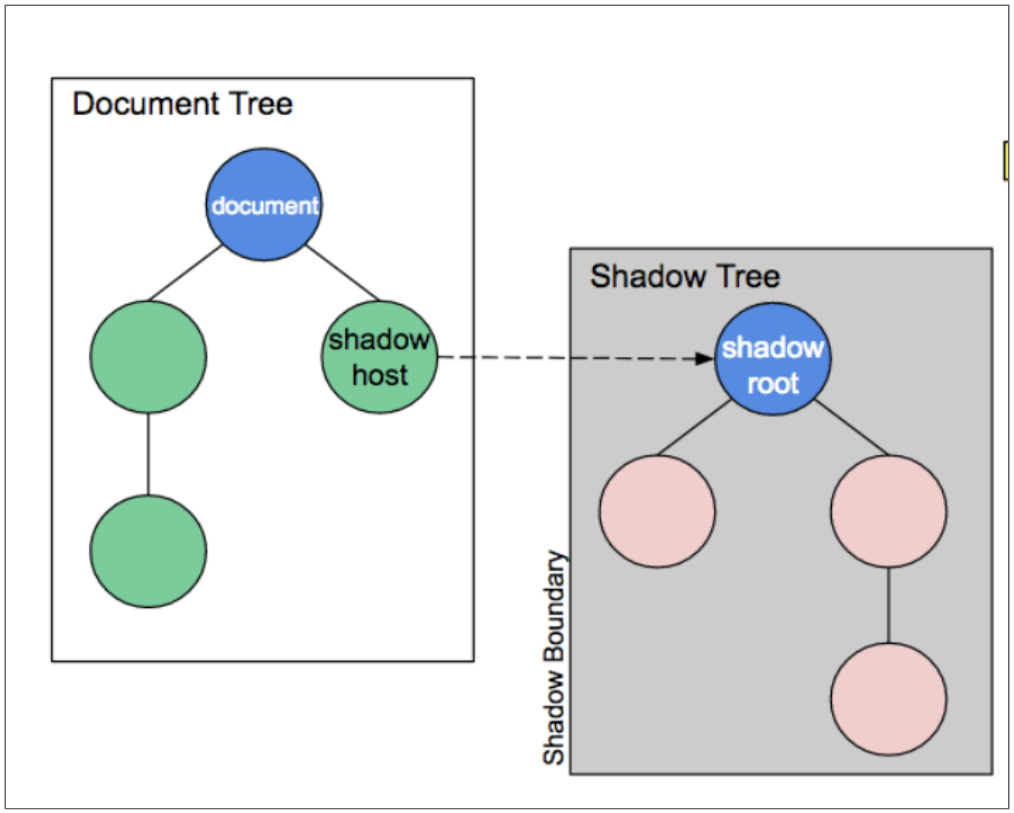
HTML TEMPLATES

```
<template id="my-paragraph">
        <p>My paragraph</p>
</template>
```

CUSTOM ELEMENTS

```
customElements.define('my-paragraph',
class extends HTMLElement {
  constructor() {
        super();
        let template = document.getElementById('my-paragraph');
        let templateContent = template.content;

        const shadowRoot = this.attachShadow({mode: 'open'})
          .appendChild(templateContent.cloneNode(true));
}
 })
```

SHADOW DOM

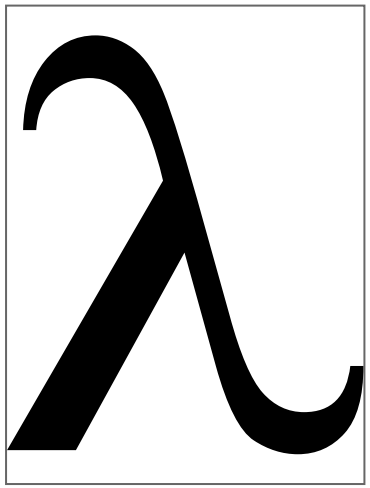


https://medium.com/javascript-in-plain-english/understanding-the-shadow-dom-8495860b39bf

Zur Kapselung von Styles und Funktionalität

FUNKTIONALE PROGRAMMIERUNG

DAS LAMBDA KALKÜL



https://de.wikipedia.org/wiki/Lambda-Kalk%C3%BClhttps://en.wikipedia.org/wiki/Alonzo_Church

KEIN NÄHERES EINGEHEN AUF DETAILS

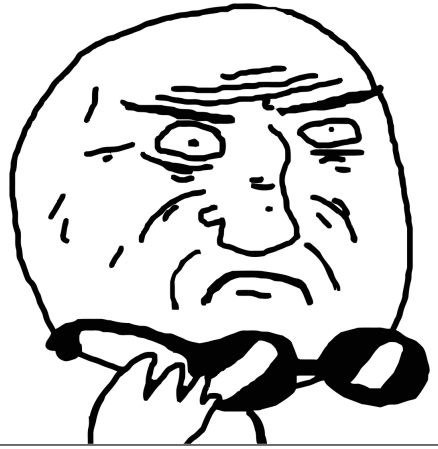
$$true = \lambda x.\ \lambda y.\ x$$

$$false = \lambda x.\lambda y.\ y$$

$$
\begin{array}{lll}
 & (\underline{\lambda f}.\lambda g.\lambda h.fg(h\,h))\underline{(\lambda x.\lambda y.x)}h(\lambda x.x\,x) & \\
\rightarrow_\beta & (\lambda g.\underline{\lambda h}.(\lambda x.\lambda y.x)g(\underline{h\,h}))h(\lambda x.x\,x) & (1)\\
\rightarrow_\alpha & (\underline{\lambda g}.\lambda k.(\lambda x.\lambda y.x)g(k\,k))\underline{h}(\lambda x.x\,x) & (2)\\
\rightarrow_\beta & (\underline{\lambda k}.(\lambda x.\lambda y.x)h(k\,k))\underline{(\lambda x.x\,x)} & (3)\\
\rightarrow_\beta & (\underline{\lambda x}.\lambda y.x)\underline{h}((\lambda x.x\,x)\ (\lambda x.x\,x)) & (4)\\
\rightarrow_\beta & (\underline{\lambda y}.h)\underline{((\lambda x.x\,x)\ (\lambda x.x\,x))} & (5)\\
\rightarrow_\beta & h & (6)
\end{array}
$$

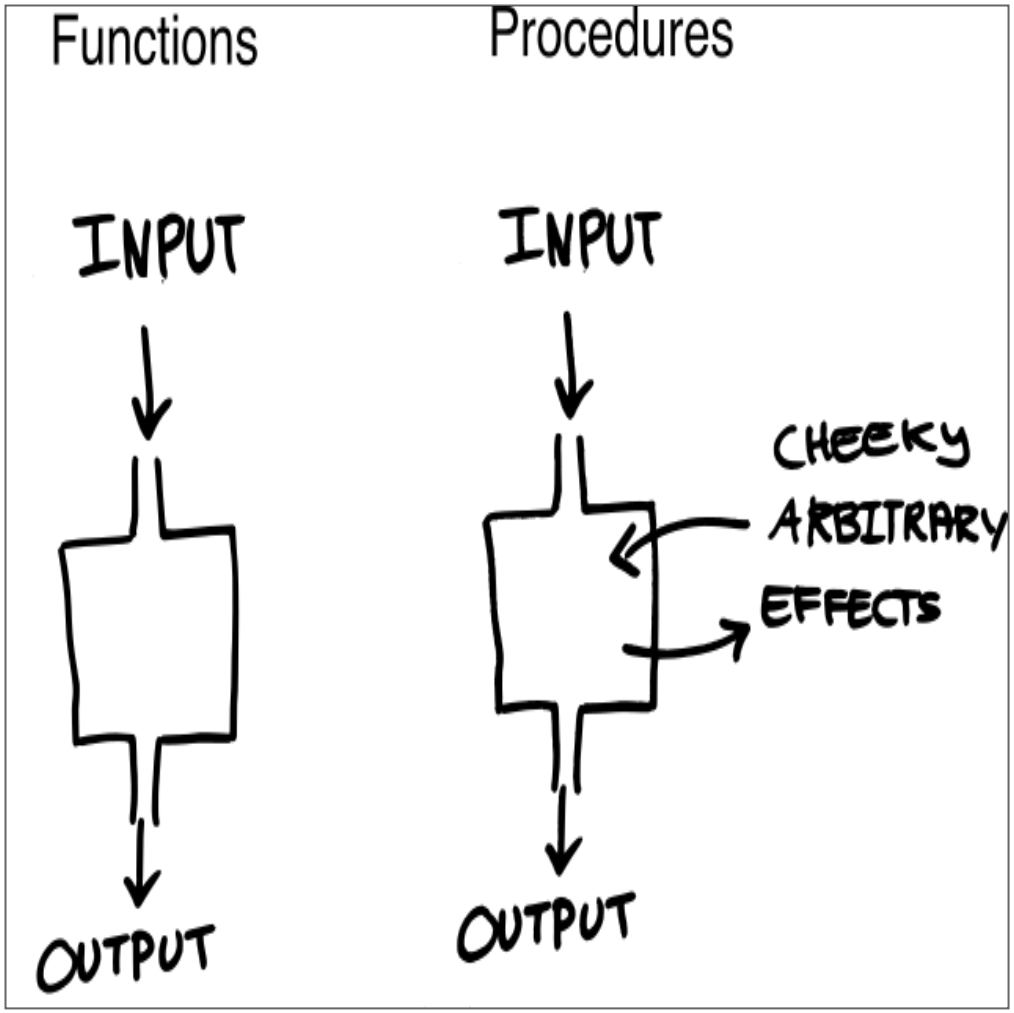
https://www.slideshare.net/MahsaSeifikar/lambda-calculus-57240548

MOTHER OF GOD...



EINE "PURE" FUNKTION

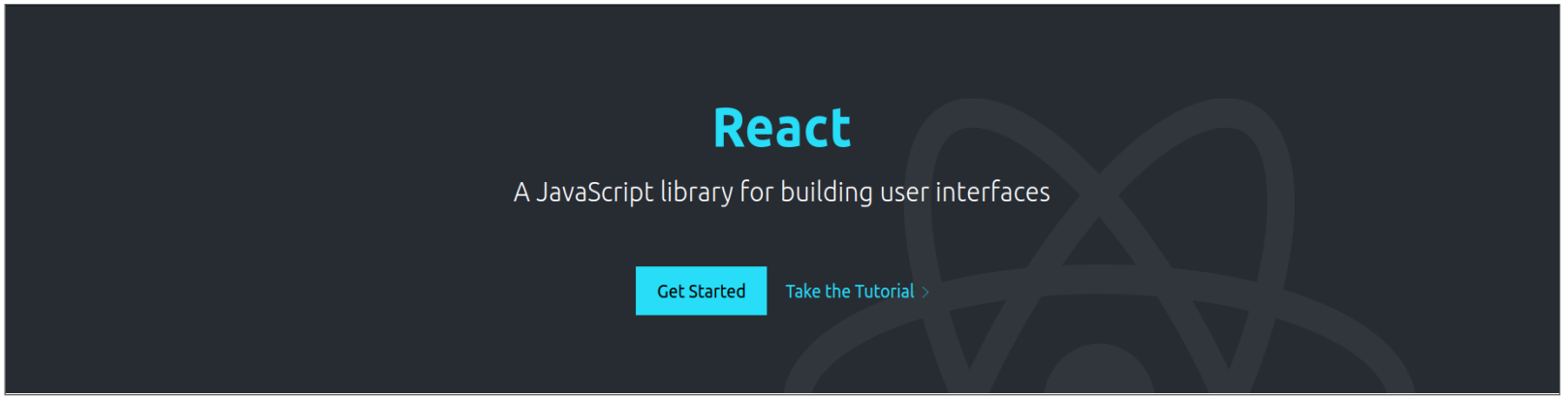
$f(x)=x+3$


Functions
Procedures
INPUT
INPUT
CHEEKY
ARBITRARY
EFFECTS
OUTPUT
OUTPUT

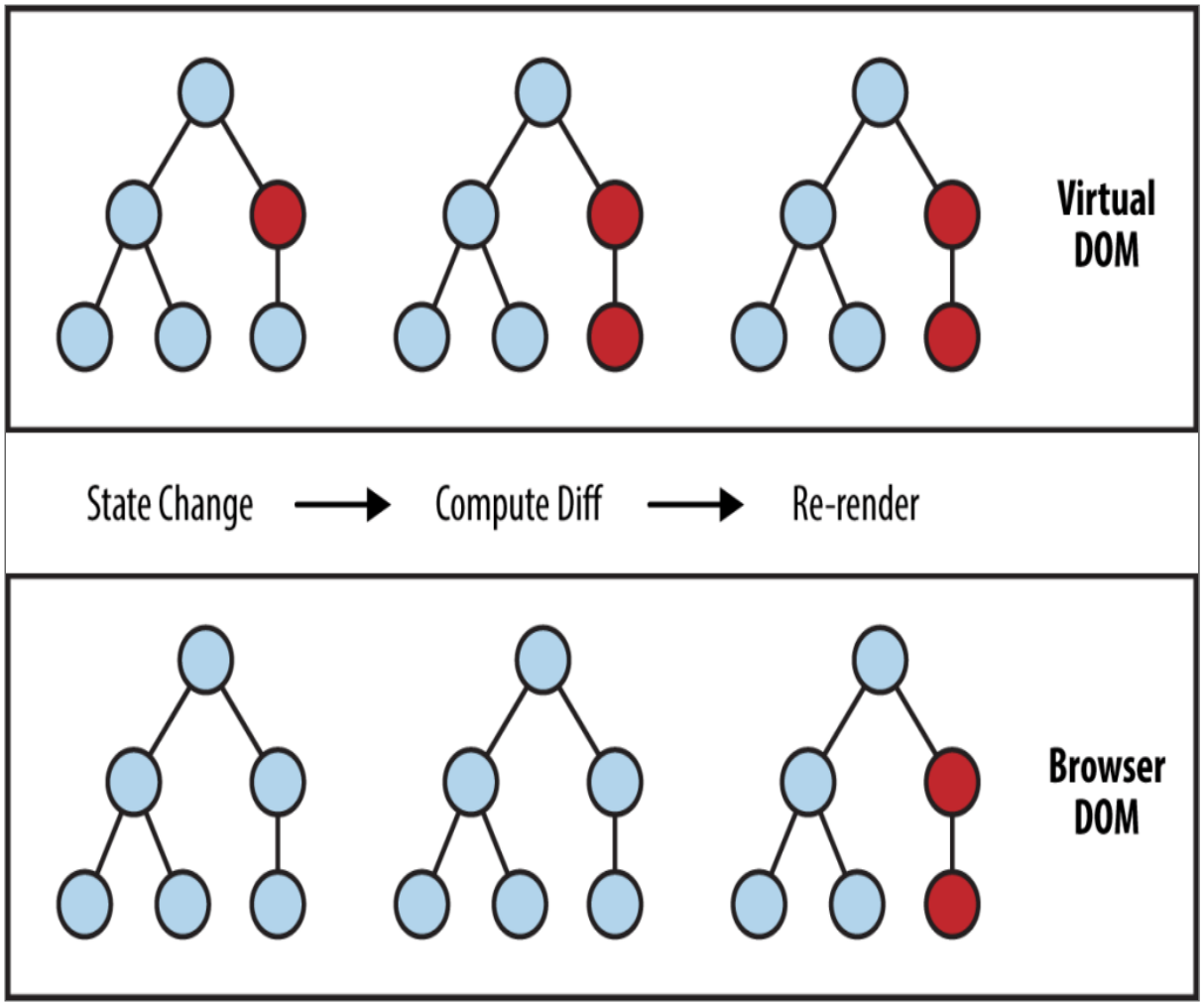
```
const fac = (n) => {
        if (n === 0) return 1;
        if (n >= 1) return n * fac(n-1);
}
```

REACT

REACT IST EIN FRAMEWORK



VIRTUAL DOM


Virtual
DOM
State Change
Compute Diff
Re-render
Browser
DOM

ERSTELLEN UND EINBINDEN VON REACT COMPONENTS

```
1   import React from "react";
2   import ReactDOM from "react-dom";
3   import "./styles.css";
4   const App = () => {
5       return (
6           <div className="App">
7           <h1>Hello CodeSandbox</h1>
8           <h2>Start editing to see some magic happen!</h2>
9           </div>
10      );
11  }
12  const rootElement = document.getElementById("root");
13  ReactDOM.render(<App />, rootElement);
```

JSX: JAVASCRIPT MIT XML

```
<?xml version="1.0"?>
- <job>
  - <production>
      <ApprovalType>WebCenter</ApprovalType>
      <Substrate>carton 150 gr</Substrate>
      <SheetSize>220-140</SheetSize>
      <press>SuperFlat2</press>
      <finishing>standard</finishing>
      <urgency>normal</urgency>
    </production>
  - <customer>
      <name>FruitCo</name>
      <number>2712</number>
      <currency>USD</currency>
    </customer>
</job>
```

ERGEBNISS IM ECHTEM DOM

PROPS, ODER AUCH DER INPUT FÜR DIE FUNKTION

```
<Counter count=0 />
```

DAS MÄCHTIGE PROP-OBJECT

```
import React, { useState } from 'react';

export const Counter = ( props ) => {

  return (
        <div>
          <p>{this.props.count}</p>
        </div>
  );
}
```

Alle datentypen in JavaScript können als Prop verwendet werden

CHILDREN, DAS BESONDERE PROP-OBJECT

```
import React from 'react';

export const Counter ( props ) => {

  return (
        <div>
          <div>{this.props.children}</div>
        </div>
  );
}
```

VERWENDUNG VON "CHILDREN"

```
<Counter>
        <ChildComponent>...</ChildComponent>
        <ChildComponent>...</ChildComponent>
        <ChildComponent>...</ChildComponent>
</Counter>
```

STATES UND CONDITIONAL RENDERING

- Props sind Immutable
- Man möchte Daten anpassen innerhalb einer Komponente
- Häufig um Styles anzupassen

CODE BEISPIEL:

```
import React, { useState } from 'react';

export const Counter = (props) => {
  const [count, setCount] = useState(this.props.count);

  return (
        <div>
          {count >= 1 ? <p>You clicked {count} times</p> : ""}
          <button onClick={() => setCount(count + 1)}>
                Click me
          </button>
        </div>
  );
}
```

WEITERE HOOKS

useEffect()

```
import React, { useEffect } from 'react';

...
useEffect(() => {
        // Update the document title using the browser API
        document.title = `You clicked ${count} times`;
  });
...
```

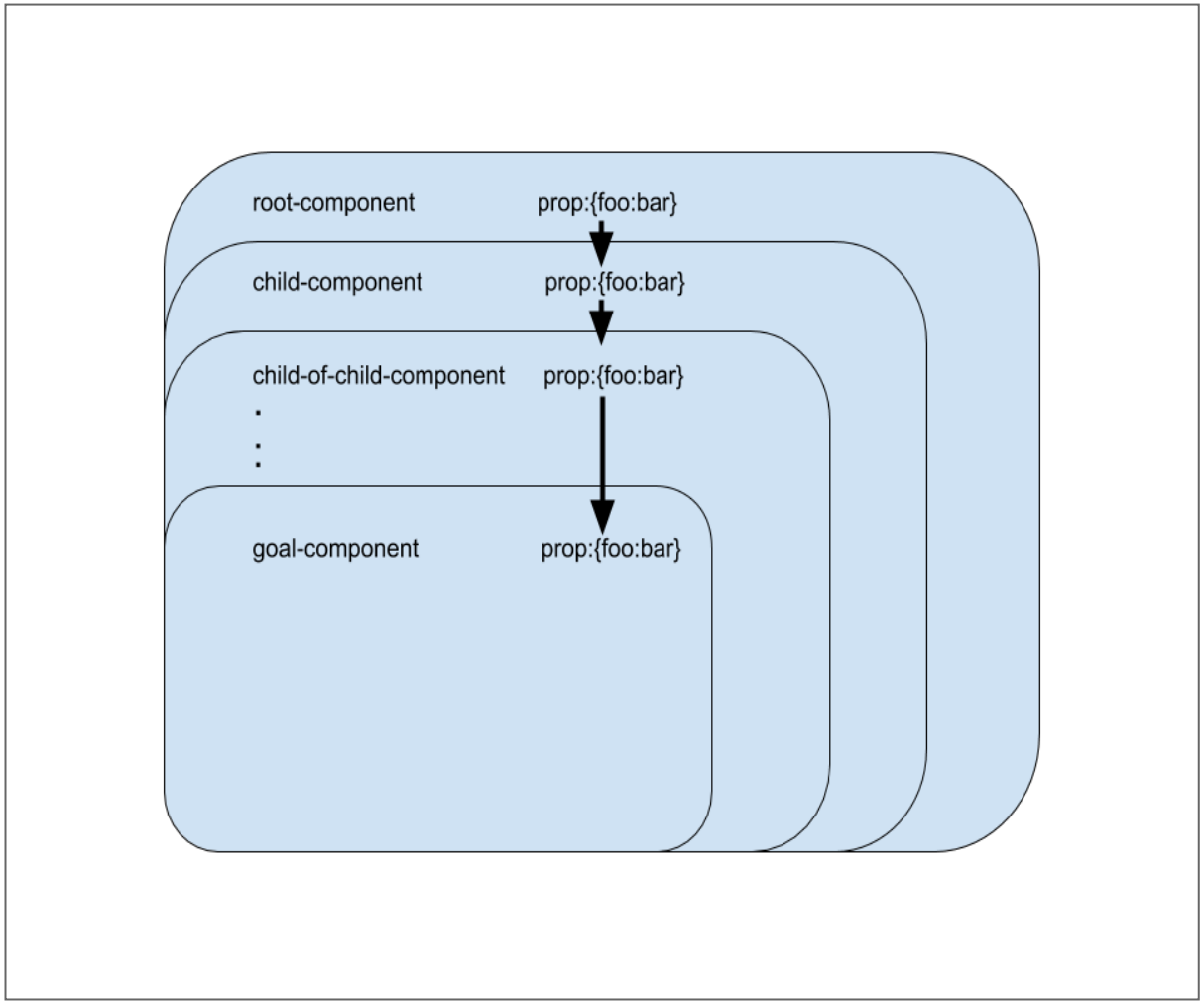
useMemo()

```
import React, { useMemo } from 'react';

...
const myMemo = useMemo(() => computeExpensiveValue(a, b), [a, b]);
...
```

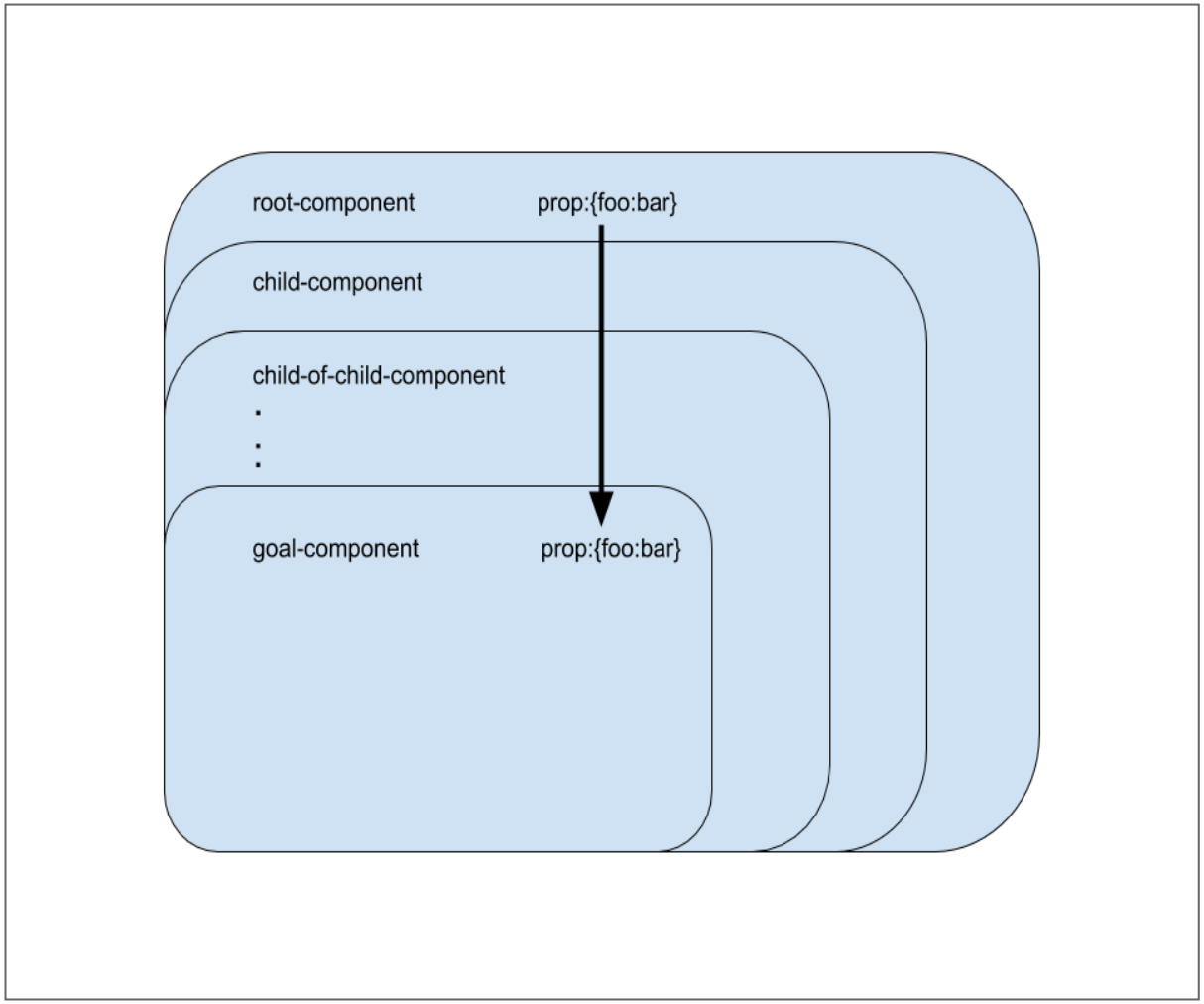
CONTEXT API

DIE RETTUNG VON PROP-DRILLING

WAS IST PROP-DRILLING?



LÖSUNG: GEZIELT PROPS AN KOMPONENTE ÜBERGEBEN



CODE BEISPIELE:

```
const themes = {
  light: {
    foreground: "#000000",
    background: "#eeeeee"
  },
  dark: {
    foreground: "#ffffff",
    background: "#222222"
  }
};

const ThemeContext = React.createContext(themes.light);

function App() {
  return (
    <ThemeContext.Provider value={themes.dark}>
      <Toolbar />
    </ThemeContext.Provider>
  );
}

function Toolbar(props) {
  return (
    <div>
      <ThemedButton />
    </div>
  );
}

function ThemedButton() {
  const theme = useContext(ThemeContext);

  return (
    <button style={{ background: theme.background, color: theme.foreground }}>
      I am styled by theme context!
    </button>
  );
}
```

WIE STYLED MAN EINE KOMPONENTE

STYLE OBJEKT IN DER COMPONENTE

```
...
        const mystyle = {
                color: "white",
                backgroundColor: "DodgerBlue",
                padding: "10px",
                fontFamily: "Arial"
          };
          return (
                <div>
                <h1 style="{mystyle}">Hello Style!</h1>
                <p>Add a little style!</p>
                </div>
          );
...
```

STYLE ÜBER EINE CSS KLASSE

Definiere ein CSS Klasse

Importiere die CSS-Datei

Verwende className statt class

```
...
import "./styles.css";
const App = () => {
        return (
                <div className="App">
                <h1>Hello CodeSandbox</h1>
                <h2>Start editing to see some magic happen!</h2>
                </div>
        );
}
...
```

ZUSAMMENGEFASST KANN MAN SAGEN:

- React ist eine Erweiterung des Web-Browsers und seiner API
- Es ist reaktive bei Änderung
- Deswegen verhält es sich eigentlich wie ein Framework
- Ist aber eine Library

DER FRUSTIERENDE TEIL

WORUM SOLL ES GEHEN?

- Die Komponente Mensch
- Overengeenering, Overrefactoring, Overeverything
- Mythen werden zu Wahrheiten
- ...es wird kein haten auf einzelne

SIE WERDEN NICHT ALLEINE ARBEITEN

- Entweder in einem Team
- Im Kontakt mit Kunden/Stakeholdern

HÄUFIG FÄLLEN KUNDEN ENTSCHEIDUNGEN, WEIL...

- ...Sie denken sie müsse die Entscheidung treffen
- ...Sie mal gehört haben dass das (Technologie) total im trend wäre
- ...Sie auf die jenigen hören die es besser verkaufen können

IHRE KOLLEGEN WERDEN...

- ...ihre eigene Karriere haben
- ...eher nach deren Meinung gefragt werden
- ...nicht auf schlüssige Argumente hören

WENN SIE NACH EINER STELLE SUCHEN, WORAUF ACHTEN SIE?

AUF XING FINDET MAN DANN HÄUFIG FOLGENDES:

- Erfahrung in [Liste von Frameworks]
- Leidenschaft für Design
- Oder, Sie sind Experte in [beliebges Framework]

> *“Wir suchen jemanden mit Erfahrung in, HTML, CSS, JavaScript, Less, Sass, Angular, React, Vue.js, Next.js, Node, XML, JSON, JWT, Drupal, CMS, WordPress, Fetch und Cypress”*

"EXPERTE"

- Anekdote: Der Koch der nur Kartoffeln kann
- 

- Quereinsteiger die nur React o.ä. gelernt haben

TECHNOLOGIE STACKS WERDEN GRÖSSER

“because frontend developers identify personally with the frameworks they use, you can’t just impose a single framework - like React or Vue - across an entire enterprise. They’ll be sharpening their pitchforks and chasing you out of town. That’s where Web Components come in. Because they can work with any framework, a central UX or design team can standardize on a single set of fully-custom, fully-functional UI components, and share them throughout their entire org, without having to persuade any teams to give up their religion.”

HTML, CSS, JavaScript

Node

React

TypeScript

Emotion (Styled Component Library)

UI-Kit Antd

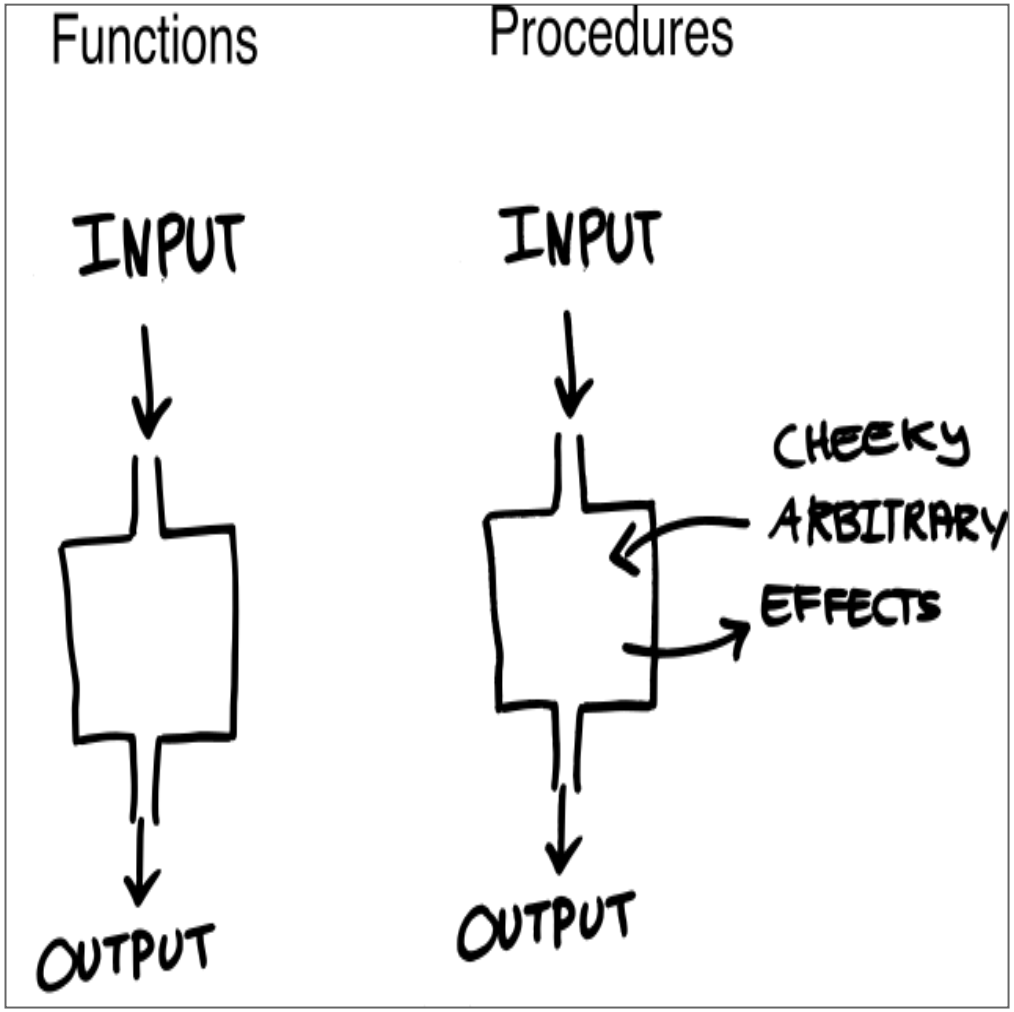
Gatsby

PROBLEME WEGEN DER KOMBIATION

MAN BAUT FIESSE HACKS

```
const ResponseButton = styled(({ children, ...props }) => (
  <Button {...props}>{children}</Button>
))(props => ({
  margin: "0.7em",
  width: "10em",
  backgroundColor: props.buttoncolor,
  color: colors.white
}));
```

ES WIRD OOP VERWENDET OBWOHL MAN FUNKTIONAL ENTWICKELN SOLLTE



```
const fac = (n) => {
        if (typeof n == string) {
                if (n < 0) {
                        if (n === 0) return 1;
                        if (n >= 1) return n * fac(n-1);
                } else {
                        return "Input should be not smaller then 0"
                }
        } else {
                return "Input should be not a string"
        }
}
```

```
const fac = (n) => {
        if (n === 0) return 1;
        if (n >= 1) return n * fac(n-1);
}
```

ARTICLE

Evaluating distributed functional languages for telecommunications software

Authors: J. H. Nyström, P. W. Trinder, D. J. King Authors Info & Affiliations

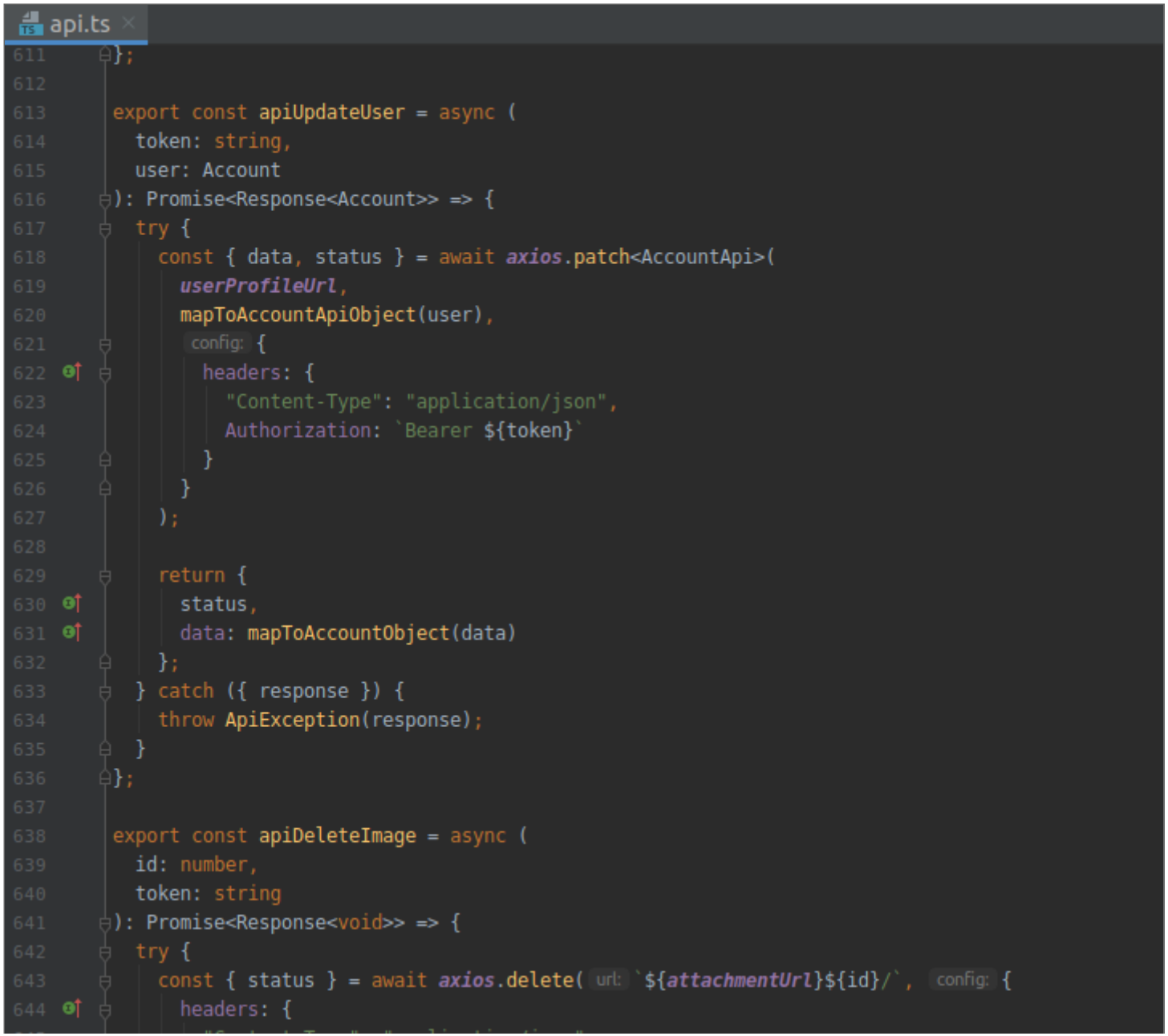
Publication: ERLANG '03: Proceedings of the 2003 ACM SIGPLAN workshop on Erlang • August 2003 • Pages 1–7 • https://doi.org/10.1145/940880.940881

ABSTRACT

The distributed telecommunications sector not only requires minimal time to market, but also software that is reliable, available, maintainable and scalable. High level programming languages have the potential to reduce development time and improve maintainability due to their compact code size. Moreover reliability is improved by safe type systems and relatively easy verification.This paper outlines plans and initial results from a joint project between Motorola and Heriot-Watt University that aims to evaluate the suitability of distributed functional languages for constructing telecommunications software. The evaluation will use the ERLANG and Glasgow distributed Haskell(GdH) languages, and be based on the construction of several typical applications. The evaluation will focus on reliability issues like ease of verification, availability issues like fault-tolerance or resilience, as well as whether the

WEITERE EIGENARTEN VON OOP

ZUM BEISPIEL: MODULARIESIERUNG VON CODE


api.ts
};
export const apiUpdateUser = async (
token: string,
user: Account
): Promise<Response<Account>> => {
try {
const { data, status } = await axios.patch<AccountApi>(
userProfileUrl,
mapToAccountApiObject(user),
config: {
headers: {
"Content-Type": "application/json",
Authorization: `Bearer ${token}`
}
}
);
return {
status,
data: mapToAccountObject(data)
};
} catch ({ response }) {
throw ApiException(response);
}
};
export const apiDeleteImage = async (
id: number,
token: string
): Promise<Response<void>> => {
try {
const { status } = await axios.delete(url: `${attachmentUrl}${id}/`, config: {
headers: {

```
645             'Content-Type': 'application/json',
646             Authorization: `Bearer ${token}`
647           }
648         });
649
650         return {
651           status,
652           data: undefined
653         };
654       } catch ({ response }) {
655         throw ApiException(response);
656       }
657     };
658
```

api.ts config.ts

```ts
// Server Error Messages
export const API_UNKNOWN_ERROR =
  "Es ist ein Fehler aufgetreten, bitte probieren Sie es erneut";
export const API_UNAUTHORIZED_ERROR =
  "Nicht autorisiert, bitte melden Sie sich an";

// Basic config
export const server = process.env.GATSBY_API_URL; // See env files
export const apiLevel = "v1/";
export const apiUrl = `${server}${apiLevel}`;
// End points
export const accommodationsUrl = `${apiUrl}accommodations/`;
export const attachmentUrl = `${apiUrl}attachment/`;
export const myAccommodationsUrl = `${apiUrl}mine/accommodations/`;
export const inquiriesUrl = `${apiUrl}accommodationinquiries/`;
export const inquiriesBindUser = `${apiUrl}accommodationinquiries/bind/`;
export const sentInquiriesUrl = `${apiUrl}sentinquiries/`;
export const accommodationRequestUrl = `${apiUrl}inquiries/`;
export const accountUrl = `${apiUrl}account/`;
export const userProfileUrl = `${accountUrl}profile/`;
export const loginUserUrl = `${accountUrl}login/`;
export const logoutUserUrl = `${accountUrl}logout/`;
export const activateAccountUrl = `${accountUrl}activate/`;
export const acceptInvitationUrl = `${accountUrl}accept_invitation/`;
export const declineInvitationUrl = `${accountUrl}decline_invitation/`;
export const resetPasswordUrl = `${accountUrl}reset/`;
export const resetPasswordConfirmUrl = `${resetPasswordUrl}confirm`;
export const getUserFromInvitationHashUrl = `${apiUrl}user-hash/`;

```

api.ts × config.ts × definitions.ts ×

```
199  export interface AccommodationInquiryItemApi
200    extends Omit<InquiryOverviewItemApi, "accommodation" | "inquiry">,
201      Partial<Pick<InquiryOverviewItemApi, "accommodation" | "inquiry">>,
202      InquiryOverviewItemAdditionalApi {}
203
204  export interface InquiryApi extends UserDataApi {
205    id?: number;
206    arrival_date: string;
207    departure_date: string;
208    number_of_persons: number;
209    additional_info?: string;
210    accommodations: number[];
211    max_price?: string;
212  }
213
214  export type SentInquiryAccommodationApi = Omit<
215    AccommodationApi,
216    "distance" | "directions" | "description" | "images"
217  >;
218
219  export interface LandlordAnswerApi {
220    message?: string;
221    price?: string;
222  }
223
224  export interface SentInquiryItemApi
225    extends Omit<
226      InquiryOverviewItemApi,
227      "accommodation" | "inquiry" | "landlord_message" | "price"
228    > {
229    accommodation: SentInquiryAccommodationApi;
230    landlord_answer: LandlordAnswerApi;
231    landlord_email: string;
232  }
```

```
234  export interface SentInquiryMainApi extends InquiryApi {
235    created_at: string;
236  }
237
238  export interface SentInquiryApi {
239    main: SentInquiryMainApi;
240    sub: SentInquiryItemApi[];
241  }
242
```

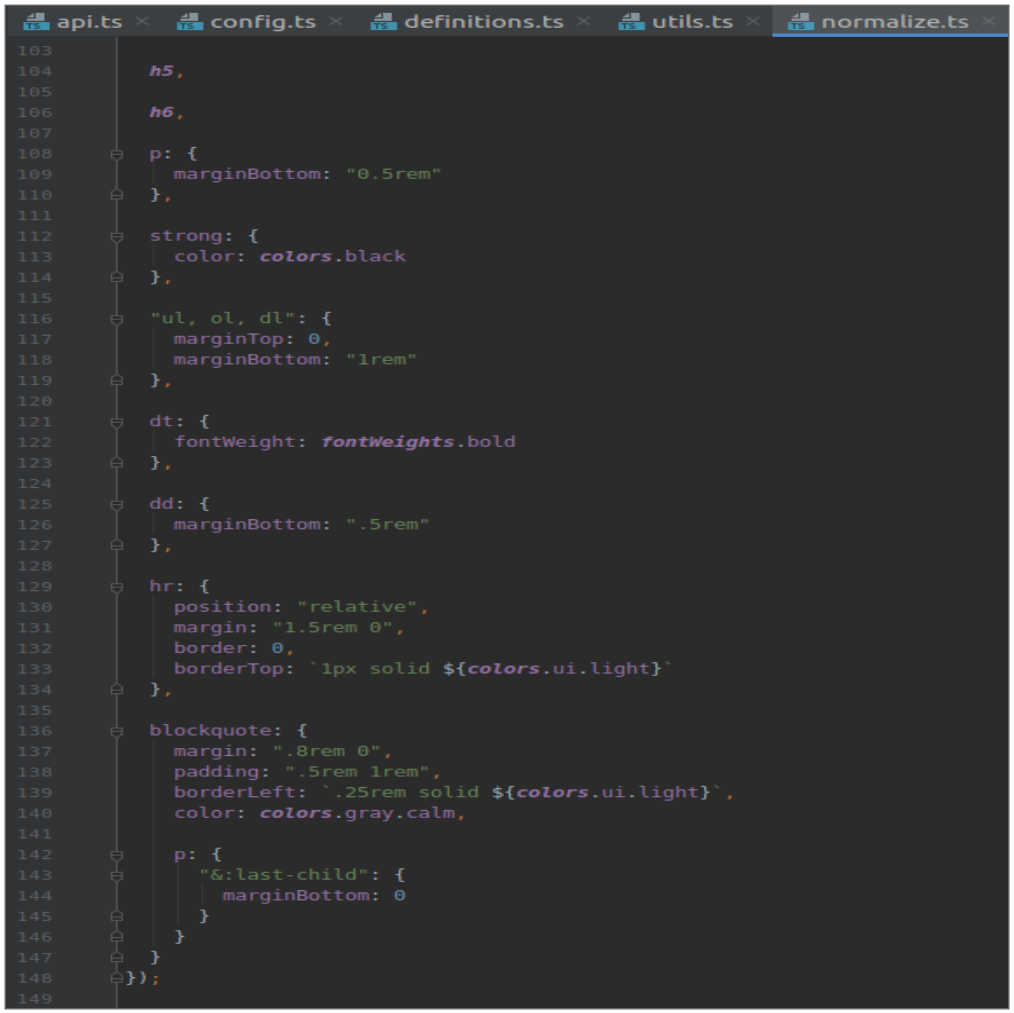

api.ts
config.ts
definitions.ts
utils.ts
500 landlord_email : string ,
501 accommodation : SentInquiryAccommodationApi ,
502 status : string ,
503 ...rest : Pick<SentInquiryItemApi, "id" | "created_at" | "number_of_persons">
504 }: SentInquiryItemApi): SentInquiryItem => ({
505 ...rest,
506 accommodation: maptoSentAccommodationObject(accommodation),
507 landlordAnswer: maptoLandlordAnswerObject(landlord_answer),
508 landlordEmail: landlord_email,
509 status: status as InquiryStatus
510 });
511
512 export const maptoSentInquiryItemApiObject = ({
513 landlordAnswer : LandlordAnswer ,
514 landlordEmail : string ,
515 accommodation : SentInquiryAccommodation ,
516 ...rest : Pick<SentInquiryItem, "id" | "status" | "createdAt" | "numberOfPersons">
517 }: SentInquiryItem): SentInquiryItemApi => ({
518 ...rest,
519 accommodation: maptoSentAccommodationApiObject(accommodation),
520 landlord_answer: maptoLandlordAnswerApiObject(landlordAnswer),
521 landlord_email: landlordEmail
522 });
523
524 export const maptoSentInquiryObject = ({
525 main : SentInquiryMainApi ,
526 sub : SentInquiryItemApi[] ,
527 ...rest
528 }: SentInquiryApi): SentInquiry => ({
529 ...rest,
530 main: maptoSentInquiryMainObject(main),
531 sub: sub.map(maptoSentInquiryItemObject)
532 });
533

```
535    main : SentInquiryMain ,
536    sub : SentInquiryItem[] ,
537    ...rest
538  }: SentInquiry): SentInquiryApi => ({
539    ...rest,
540    main: maptoSentInquiryMainApiObject(main),
541    sub: sub.map(maptoSentInquiryItemApiObject)
542  });
543
```

UND JETZT, WAS GANZ BESONDERES

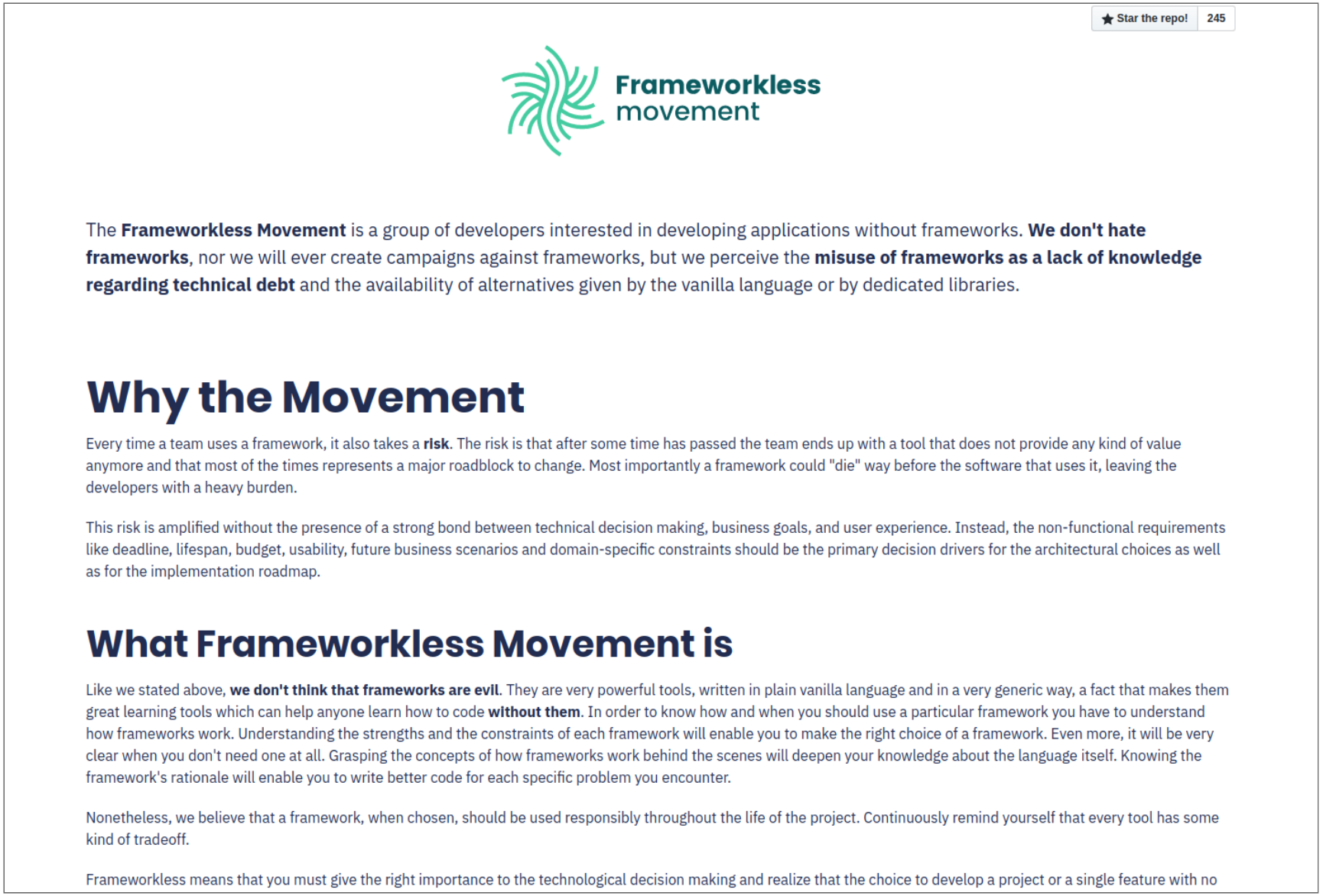
- Emotion für Styled Components
- Verwendet die Objekt-Notation damit man Typscript verwenden kann
- Erkennt das man in vielen Kompoenten die gleichen Styled Components verwendet

WAS HAT MAN ALSO GEMACHT?

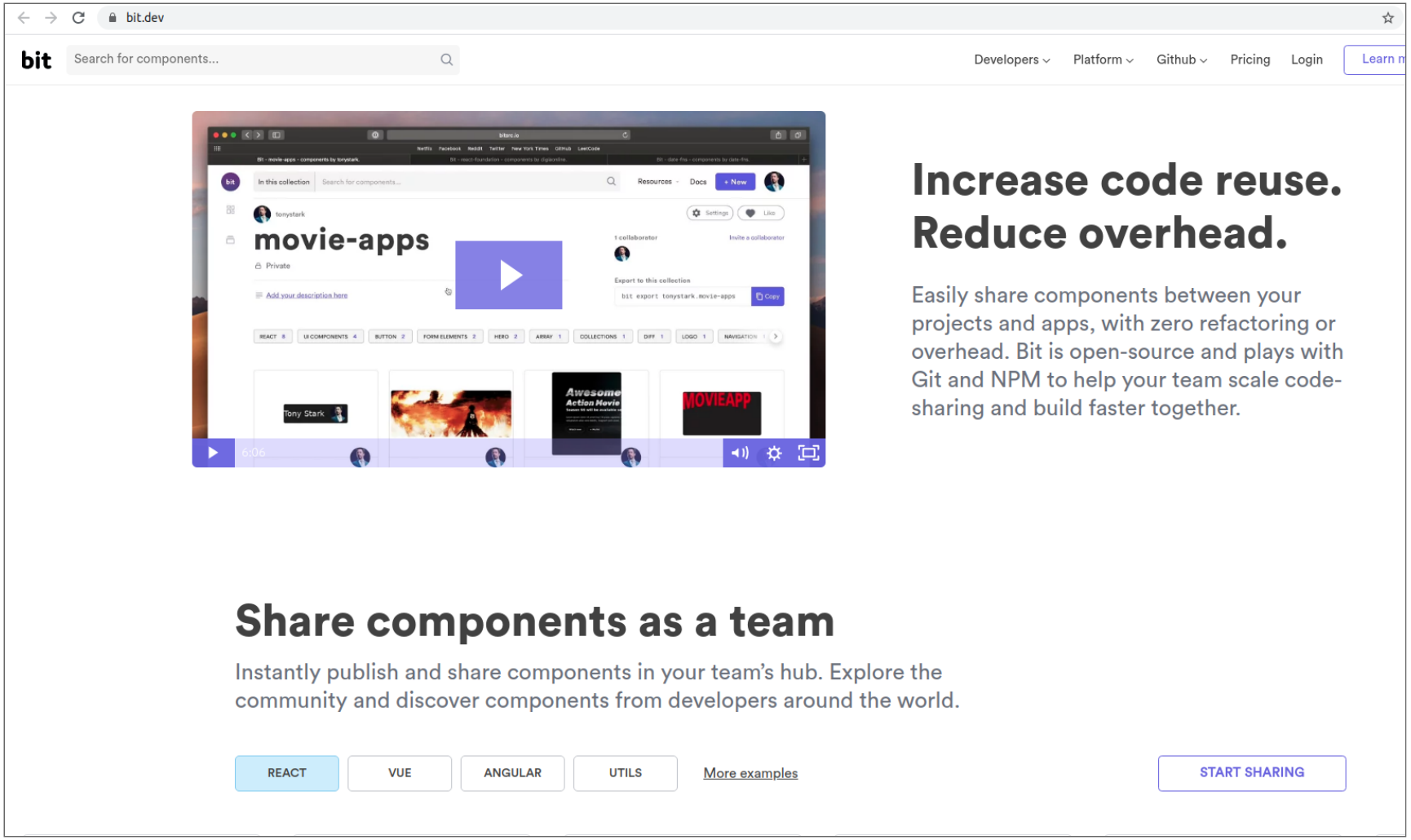


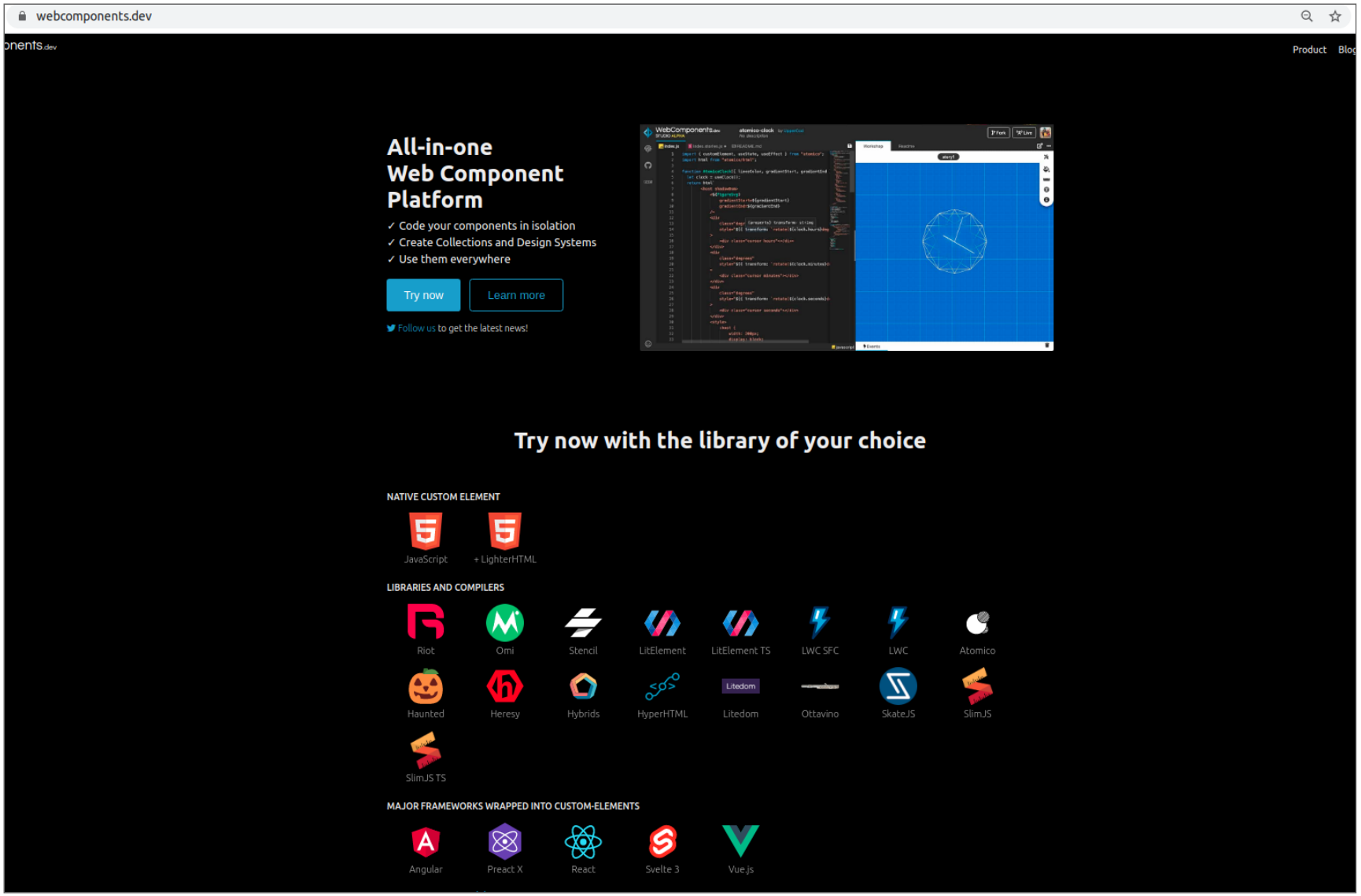

CSS
This is brilliant.
JS AS CSS
But I like this.
imgflip.com

DER TREND GEHT ZU WENIGER FRAMEWORK MEHR VANILLA


Star the repo! 245
Frameworkless movement
The Frameworkless Movement is a group of developers interested in developing applications without frameworks. We don't hate frameworks, nor we will ever create campaigns against frameworks, but we perceive the misuse of frameworks as a lack of knowledge regarding technical debt and the availability of alternatives given by the vanilla language or by dedicated libraries.
Why the Movement
Every time a team uses a framework, it also takes a risk. The risk is that after some time has passed the team ends up with a tool that does not provide any kind of value anymore and that most of the times represents a major roadblock to change. Most importantly a framework could "die" way before the software that uses it, leaving the developers with a heavy burden.
This risk is amplified without the presence of a strong bond between technical decision making, business goals, and user experience. Instead, the non-functional requirements like deadline, lifespan, budget, usability, future business scenarios and domain-specific constraints should be the primary decision drivers for the architectural choices as well as for the implementation roadmap.
What Frameworkless Movement is
Like we stated above, we don't think that frameworks are evil. They are very powerful tools, written in plain vanilla language and in a very generic way, a fact that makes them great learning tools which can help anyone learn how to code without them. In order to know how and when you should use a particular framework you have to understand how frameworks work. Understanding the strengths and the constraints of each framework will enable you to make the right choice of a framework. Even more, it will be very clear when you don't need one at all. Grasping the concepts of how frameworks work behind the scenes will deepen your knowledge about the language itself. Knowing the framework's rationale will enable you to write better code for each specific problem you encounter.
Nonetheless, we believe that a framework, when chosen, should be used responsibly throughout the life of the project. Continuously remind yourself that every tool has some kind of tradeoff.
Frameworkless means that you must give the right importance to the technological decision making and realize that the choice to develop a project or a single feature with no


OK looks like a simple script will do – we can do this in vanilla JS, no problem!
No need for a framework...
Hmm, I guess I should make this a bit more object oriented...
...and it can't hurt to separate out the front-end with templates...
And the code will be cleaner if I add an abstraction layer over the database and if I use API endpoints
I guess I could document it and put it on Github, maybe it'll help someone out...
WHAT'S THIS?? ANOTHER Javascript Framework!? And already 1000 stars on Github?? Tell me you didn't...
I mean... it wasn't on purpose...
CommitStrip.com


bit.dev
bit
Search for components...
Developers
Platform
Github
Pricing
Login
Learn m
movie-apps
Increase code reuse.
Reduce overhead.
Easily share components between your projects and apps, with zero refactoring or overhead. Bit is open-source and plays with Git and NPM to help your team scale code-sharing and build faster together.
Share components as a team
Instantly publish and share components in your team's hub. Explore the community and discover components from developers around the world.
REACT
VUE
ANGULAR
UTILS
More examples
START SHARING


webcomponents.dev
Product
All-in-one
Web Component
Platform
✓ Code your components in isolation
✓ Create Collections and Design Systems
✓ Use them everywhere
Try now
Learn more
Follow us to get the latest news!
Try now with the library of your choice
NATIVE CUSTOM ELEMENT
JavaScript
+ LighterHTML
LIBRARIES AND COMPILERS
Riot
Omi
Stencil
LitElement
LitElement TS
LWC SFC
LWC
Atomico
Haunted
Heresy
Hybrids
HyperHTML
Litedom
Ottavino
SkateJS
SlimJS
SlimJS TS
MAJOR FRAMEWORKS WRAPPED INTO CUSTOM-ELEMENTS
Angular
Preact X
React
Svelte 3
Vue.js

blog.bitsrc.io/11-must-know-frontend-trends-for-2020-cea8a629b08

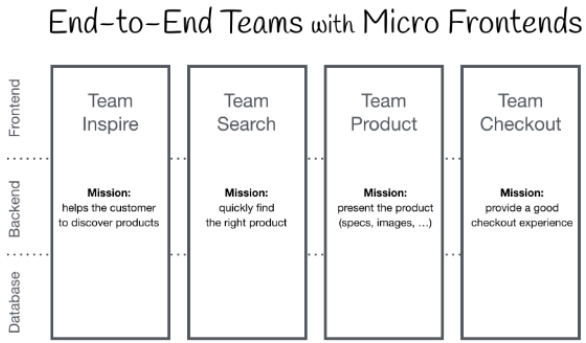
Enjoy!

. . .

1. Micro frontends

Micro Frontends are the buzziest frontend topic for lunch conversations.

Ironically, while frontend development enjoys the modular advantages of components, it is still largely more monolithic than backend microservices.



Micro frontends bring the promise of splitting your frontend architecture into different frontends for different teams working on different parts of your app. Each team can gain autonomy over the end-to-end lifecycle of their micro frontend, which can be developed, versioned, tested, built

Bits and Pieces
Scale frontend • Build and deploy components • Compose UIs together • https://bit.dev ⚡

Follow

jaxenter.de/micro-frontends-angular-elements-web-components-steyer-89889

JAXenter NEWS ARTIKEL VIDEOS JAVA MAGAZIN E.ACADEMY JAX 2020 SUCHE

Java DevOps Microservices Serverless Kubernetes Blockchain Eclipse Android Agile Machine Learning

Quickvote: Womit befassen Sie sich 2020?

DevOpsCon 2020 | 23. bis 26. März, Singapur

Best of 2019 Kiosk Special Vol. 1: Technologien, die Sie kennen sollten!

JAX 2020 | 11. bis 15. Mai, Mainz

Manfred Steyers Session auf der W-JAX 2019

Micro Frontends mit Angular Elements und Web Components – eine perfekte Kombination?

#Angular #W-JAX 2019 #Web Components

5. Dezember 2019 Redaktion JAXenter

Die Idee von Micro Frontends ist sehr verlockend: Anstatt eines großen monolithischen Clients erstellt man entsprechend der Microservices-Philosophie mehrere kleine und gut wartbare UIs. Doch wie lassen sich diese einzelnen Inseln mit einem integrierten UI präsentieren? Framework-unabhängige Web Components, die sich dynamisch laden lassen, ermöglichen hier gleich mehrere attraktive Lösungsansätze. Wieso das so ist, erklärt Manfred Steyer (SOFTWAREarchitekt.at) in seiner Session auf der W-JAX 2019.

In seiner Session auf der W-JAX 2019 erklärt Manfred Steyer, wie Sie diese Idee mit Angular Elements umsetzen können und wie Sie durch den Einsatz des neuen Ivy-Compilers praxistaugliche Bundle-Größen für Ihre Web Components erreichen. Darauf aufbauend besprechen wir mehrere Umsetzungsstrategien und diskutieren die Vor- und Nachteile. Am Ende sind Sie in der Lage, die einzelnen Optionen für Ihre Vorhaben zu bewerten.

entwickler.kiosk
Best of 2019
Technologien, die Sie kennen sollten!

POLLS

Womit befassen Sie sich 2020?

- Update auf eine neue Java-Version
- Enterprise Java mit Jakarta EE
- Java im Container (Kubernetes / Docker / etc...)
- Cloud Computing mit Java
- Java als UI-Technologie: JavaFX
- Entwicklung im Mobile-Bereich
- Big-Data-Technologien
- Neuerungen im Tooling-Bereich (Eclipse, NetBeans, Git, Hudson/Jenkins, Maven, etc.)
- Neuerungen bei anderen etablierten JVM-Sprachen wie Groovy, Scala, Clojure, JRuby

VIELE FIRMEN SIND SCHON AUF DEM TREND

- Karstadt Galeria Kaufhof
- Rewe Digital
- Douglas

WIESO WERDEN NOCH FEHLENTSCHEIDUNGEN GETROFFEN?

Die meisten Online Materialien sind zu Oberflächlich

Die meisten Verbreiten Mythen und gefährliche Behauptungen

Learn JavaScript - Full Course for Beginners

freeCodeCamp.org 2,5 Mio. Aufrufe • vor 12 Monaten

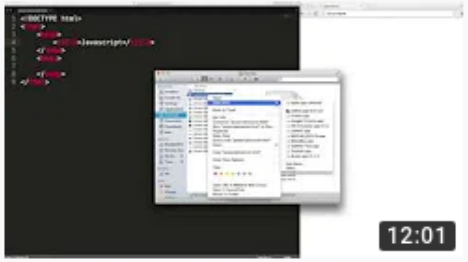
This complete 134-part **JavaScript** tutorial for beginners will teach you everything you need to know to get started with the ...

JavaScript Tutorial for Beginners: Learn JavaScript in 1 Hour [2019]

Programming with Mosh 1,6 Mio. Aufrufe • vor 1 Jahr

Watch this JavaScript tutorial for beginners to **learn JavaScript** basics in one hour. Get my Complete JavaScript Course to master ...

Untertitel



Learn JavaScript in 12 Minutes

Jake Wright • 2,1 Mio. Aufrufe • vor 5 Jahren

Learn the fundamental features of **JavaScript** - the language used to add dynamic, interactive content to websites. I teach you how ...

JavaScript Tutorial for Beginners - Full Course in 8 Hours [2020]

Clever Programmer 197.641 Aufrufe • vor 2 Monaten

Here is the best free javascript programming course on the planet. Made with lots of ❤. Take your web development skills to ...

JavaScript Tutorial for Beginners

Telusko

#0 JavaScript Tutorial | Introduction • 2:44

#1 JavaScript Tutorial | Why you should Learn JavaScript Today • 6:48

KOMPLETTE PLAYLIST ANSEHEN

Learn JavaScript in 7 minutes | Create Interactive Websites | Code in 5

blondiebytes • 310.772 Aufrufe • vor 3 Jahren

Learn how to make your websites interactive with the **JavaScript in** 7 minutes! We'll review some basic HTML/CSS concepts as ...

Computer Science Degree vs Coding Bootcamps vs Self Taught (2018 - 2019)

ENGINEERED TRUTH 36.906 Aufrufe • vor 1 Jahr

When it comes between you getting a 4-year **computer science degree**, specializing in a high demand skill by attending a coding ...

Coding Bootcamp vs Computer Science Degree | Career Health

Senegoddess • 89.034 Aufrufe • vor 2 Monaten

Hi friend! This is a question I get asked a LOT: doing a Coding **Bootcamp** versus going to college. I hope this can clear up some ...

Coding Bootcamp vs Self-Taught vs Computer Science Degree

Coder Foundry • 6163 Aufrufe • vor 6 Monaten

There are three main learning paths to becoming a coding career - **bootcamps**, college, and online courses. Most articles pit one ...

Coding Bootcamp vs Computer Science degree WHAT SHOULD YOU PICK?

Program With Erik • 1443 Aufrufe • vor 1 Jahr

Should you get a full stack certificate or persue a coding bootcamp or get a computer science degree? How do you get into IT ...

CS Degree vs Coding Bootcamp (2019) - by Rubén Harris, CEO of #CareerKarma

Career Karma • 3344 Aufrufe • vor 4 Monaten

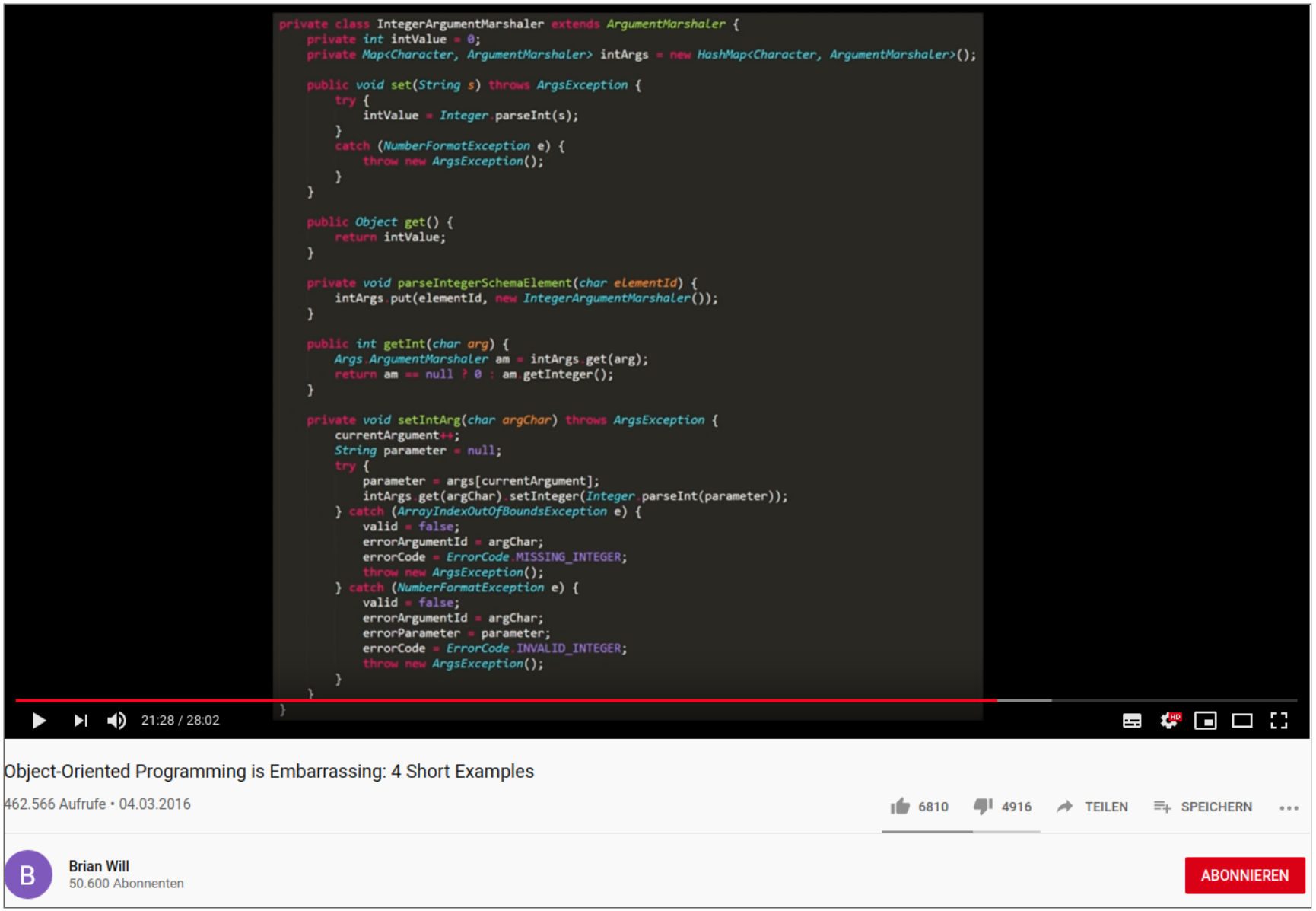
If you're here, your brains are about to explode because you're not sure about a **CS Degree vs**. Coding **Bootcamp**. Well, if you ...

CODING BOOTCAMPS: What is it and should you go to one?!

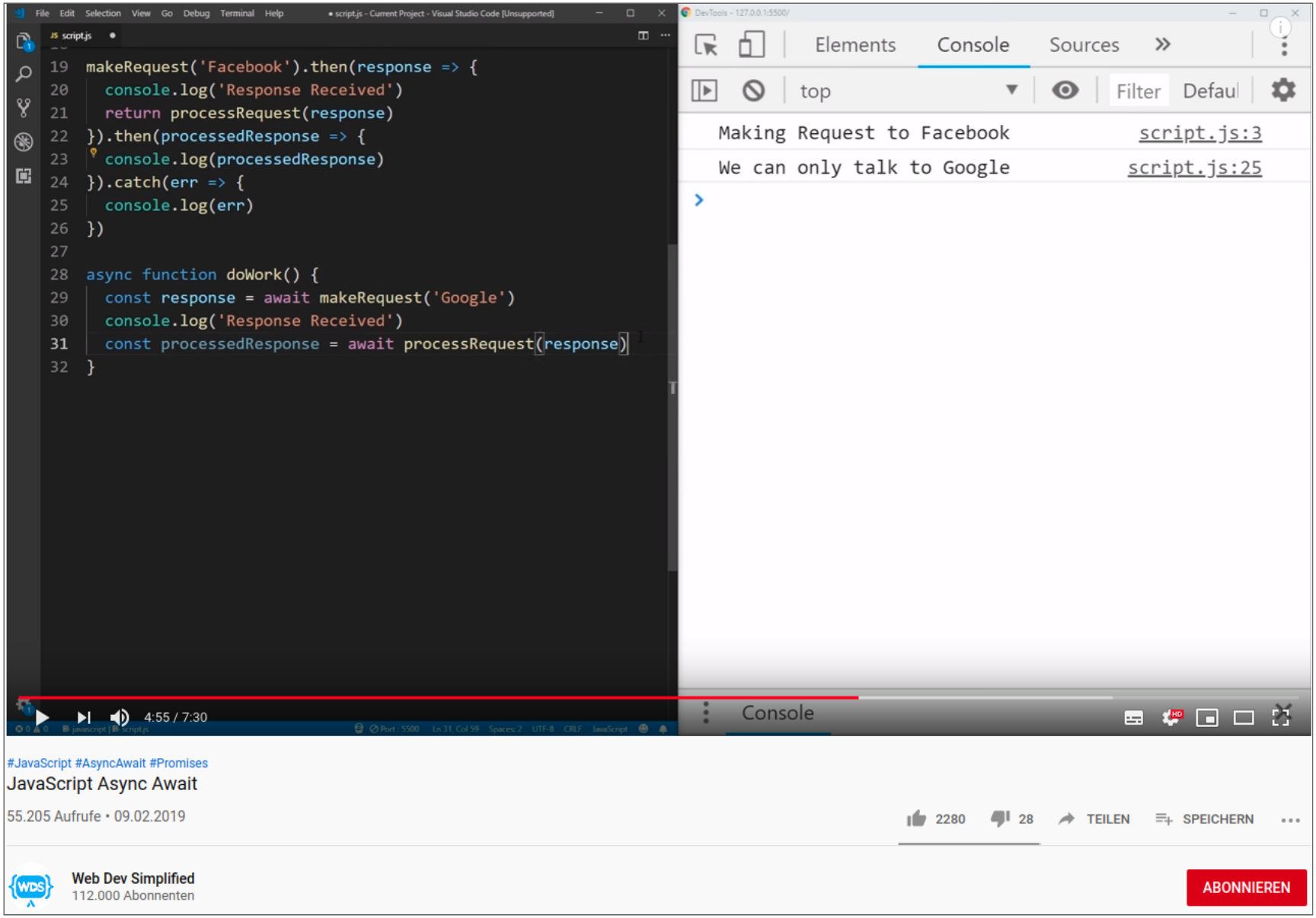
mayuko 157.596 Aufrufe • vor 1 Jahr

Hellooo everybody! Signup for your FREE trial to The Great Courses Plus here: http://ow.ly/TiIA30iEoPL Today I'm talking ...

Untertitel




```
private class IntegerArgumentMarshaler extends ArgumentMarshaler {
    private int intValue = 0;
    private Map<Character, ArgumentMarshaler> intArgs = new HashMap<Character, ArgumentMarshaler>();

    public void set(String s) throws ArgsException {
        try {
            intValue = Integer.parseInt(s);
        }
        catch (NumberFormatException e) {
            throw new ArgsException();
        }
    }

    public Object get() {
        return intValue;
    }

    private void parseIntegerSchemaElement(char elementId) {
        intArgs.put(elementId, new IntegerArgumentMarshaler());
    }

    public int getInt(char arg) {
        Args.ArgumentMarshaler am = intArgs.get(arg);
        return am == null ? 0 : am.getInteger();
    }

    private void setIntArg(char argChar) throws ArgsException {
        currentArgument++;
        String parameter = null;
        try {
            parameter = args[currentArgument];
            intArgs.get(argChar).setInteger(Integer.parseInt(parameter));
        } catch (ArrayIndexOutOfBoundsException e) {
            valid = false;
            errorArgumentId = argChar;
            errorCode = ErrorCode.MISSING_INTEGER;
            throw new ArgsException();
        } catch (NumberFormatException e) {
            valid = false;
            errorArgumentId = argChar;
            errorParameter = parameter;
            errorCode = ErrorCode.INVALID_INTEGER;
            throw new ArgsException();
        }
    }
}
```


File Edit Selection View Go Debug Terminal Help
script.js - Current Project - Visual Studio Code [Unsupported]
script.js
19 makeRequest('Facebook').then(response => {
20 console.log('Response Received')
21 return processRequest(response)
22 }).then(processedResponse => {
23 console.log(processedResponse)
24 }).catch(err => {
25 console.log(err)
26 })
27
28 async function doWork() {
29 const response = await makeRequest('Google')
30 console.log('Response Received')
31 const processedResponse = await processRequest(response)
32 }
4:55 / 7:30
Elements Console Sources
top
Filter
Making Request to Facebook script.js:3
We can only talk to Google script.js:25
Console
#JavaScript #AsyncAwait #Promises
JavaScript Async Await
55.205 Aufrufe • 09.02.2019
2280
28
TEILEN
SPEICHERN
Web Dev Simplified
112.000 Abonnenten
ABONNIEREN

KOOPERATION IM BEZUG AUF PRAXIS-PROJEKTE ODER BACHELOR-ARBEITEN

Mögliche Themen Praxis-Projekt:

- Modularisierung von Web-Componenten
- Erstellung eines Vertriebstool

MÖGLICHE THEMEN BACHELOR-ARBEIT:

- Entwicklung und Evaluierung eines Konzepts für Agenturen für Micro-Frontends
- Konzeptzonierung von Micro-Services mit gRPC

Kontaktieren Sie mich: artur.zimmermann@ambient-innovation.com

PRAKTISCHE ÜBUNG: SUPERHELDEN LISTE

ANFORDERUNGEN

- Eine Startseite mit Bild und Login (Username, Password, Login-Button)
- Username soll der Name eines Superhelden sein
- Password soll mindesten 6 Zeichen lang sein und ein Sonderzeichen haben.
- Bei erfolgreichem Login kommt man auf eine neue Seite
- Auf der nächsten Seite; Die Helden als Liste darstellen (Name, Beschreibung, Bild)

Codesandbox link: **https://codesandbox.io/s/hbrs-webengineering2020-6pd52**