

# Dyetoools

Projekt 1



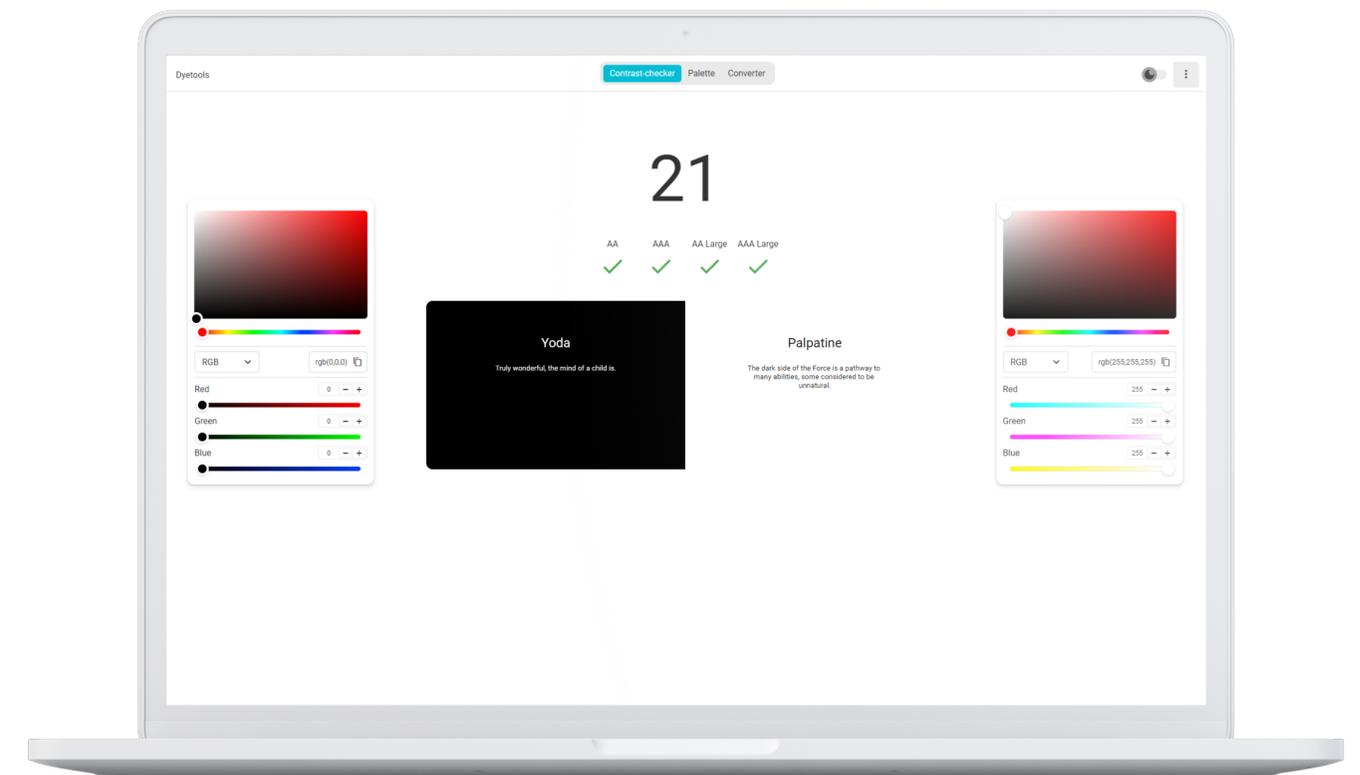
Betreut durch  
Prof. Dr. Marius Hofmeister

Unterstützt durch  
Prof. Jürgen Graef M.A.

Von  
Marc Zinser, Nils Blessing

Dyetoool ist eine Webanwendung entwickelt für Designer. Mit mehreren Tools rund um das Thema Farbe sollte eine Plattform geschaffen werde, mit der der Farbfindungs- und Überprüfungszyklus vereinfacht wird.

Neben einem Contrast Checker können ausgewählte Farben konvertiert und in Nuancen verfeinert werden. Zusätzlich bietet ein Palettengenerator die Möglichkeit, auf Basis vordefinierter Kontraste bis zu 10 komplementäre Farben bereitzustellen. In Hinsicht auf kollaborative Arbeitsweisen wurde außerdem darauf geachtet, dass mit jedem individuellen Link auch die selbe individuelle Palette angezeigt wird.





Express

chroma.js

Für die Umsetzung des Projekts wollten wir uns mit einer für uns neuen Webtechnologie entscheiden. Um evtl. mit SSR arbeiten zu können, entschieden wir uns für Nuxt.js, einer Abwandlung von Vue.js, die sowohl mit Vuex-Middleware als auch optionalem Server Side Rendering kommt.

Um größere JSON-Dateien nicht zwanghaft beim laden der Seite zu lesen, haben wir außerdem eine API eingebaut, die besagte Dateien nur laden, wenn diese gebraucht werden. Hierfür wurde Express.js genutzt.

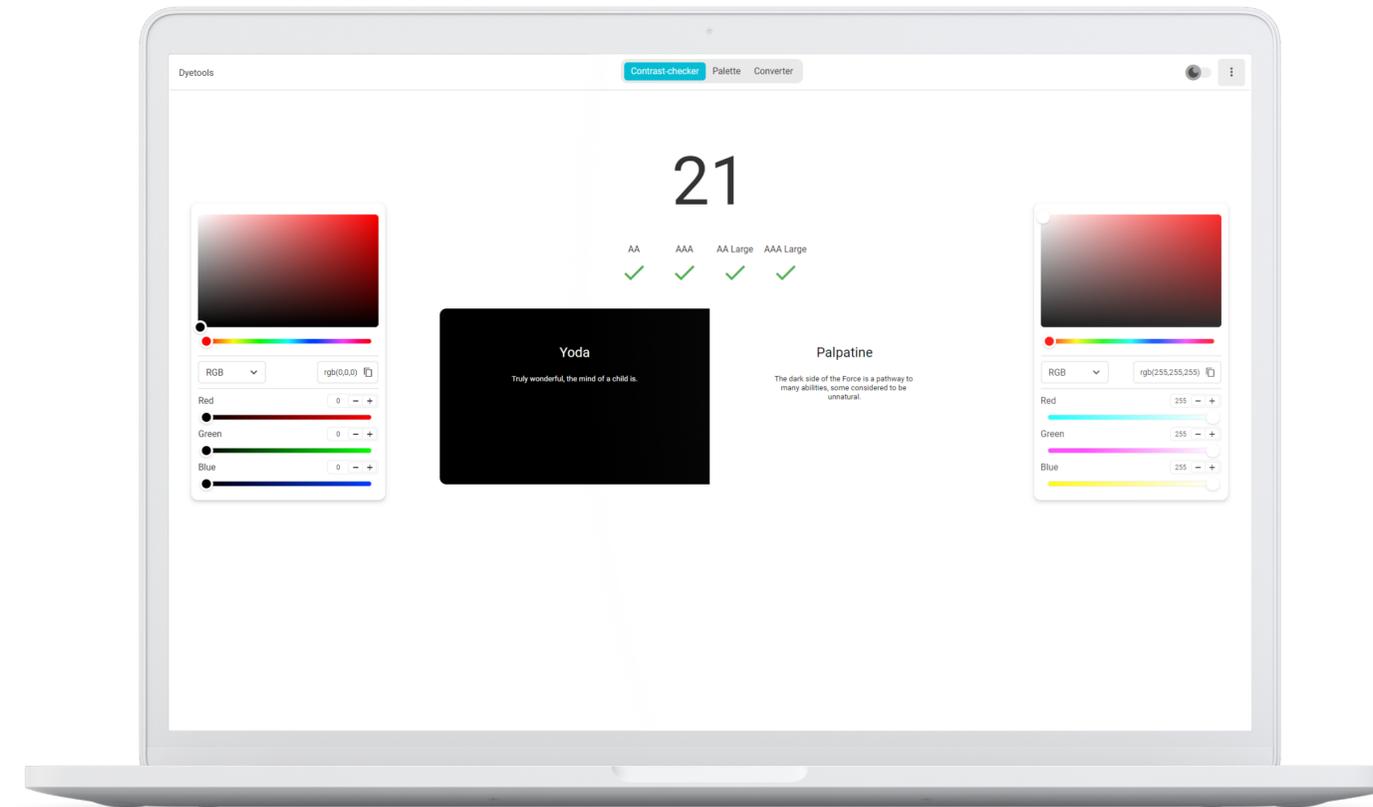
Da extrem oft Farben konvertieren mussten, haben wir chroma.js als tragendes Package für unser Projekt integriert.

Das größte Hindernis im Projekt war der custom Color Picker. Dieser besteht aus X Komponenten, die zum einen Teil funktioneller Natur sind, zum anderen erlauben, zwischen verschiedenen Farbmodi zu wechseln.

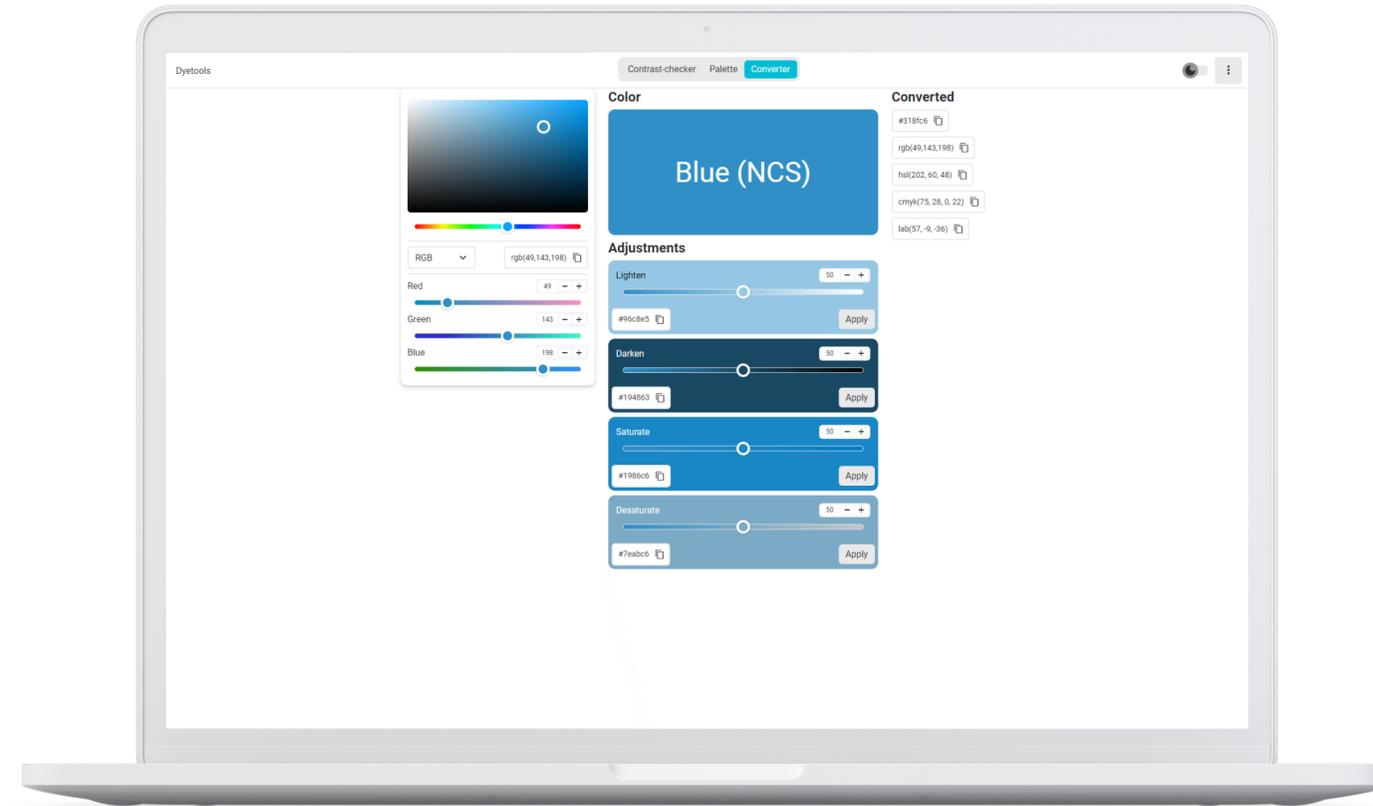
Ein weiteres Problem lag in der JSON-API, die wir für die Bereitstellung der Zitate für den Contrast Checker und das Parsen der Dateien mit den Farbdaten (Pantone, HKS, etc...)

```
1 getSliderCollection() {
2     let chrome = chroma({h: this.hue, s: this.saturation, v: this.value});
3     switch (this.sliderModes[this.activeMode].toLowerCase()) {
4         case "rgb":
5             return (<RGBSliderCollection redIn={chrome.get("rgb.r")} greenIn={chrome.get("rgb.g")} blueIn={chrome.get("rgb.b")} style="width: 100%" v-on:onChanged={this.sliderChanged} v-on:onChangeEnd={this.changeEnd}/>);
6         case "hsl":
7             return (<HSLSliderCollection hueIn={this.hue} saturationIn={chrome.get("hsl.s") * 100} lightnessIn={chrome.get("hsl.l") * 100} style="width: 100%" v-on:onChanged={this.sliderChanged} v-on:onChangeEnd={this.changeEnd}/>);
8         case "cmyk":
9             return (<CMYKSliderCollection cyanIn={chrome.get("cmyk.c") * 100} magentaIn={chrome.get("cmyk.m") * 100} yellowIn={chrome.get("cmyk.y") * 100} keyIn={chrome.get("cmyk.k") * 100} style="width: 100%" v-on:onChanged={this.sliderChanged} v-on:onChangeEnd={this.changeEnd}/>);
10        case "lab":
11            return (<LABSliderCollection lIn={chrome.get("lab.l")} aIn={chrome.get("lab.a")} bIn={chrome.get("lab.b")} style="width: 100%" v-on:onChanged={this.sliderChanged} v-on:onChangeEnd={this.changeEnd}/>);
12        case "hex":
13            return (<TextInput valid={this.hexBoxValid} placeholder="#000000" textIn={chrome.hex().toUpperCase()} v-on:changed={this.textIn}/>)
14        case "copic":
15            return (<Swatches hexIn={chrome.hex()} triggerSort={this.triggerSort} typ="copic" v-on:changed={this.textIn}/>);
16        case "ral":
17            return (<Swatches hexIn={chrome.hex()} triggerSort={this.triggerSort} typ="ral" v-on:changed={this.textIn}/>);
18        case "hks":
19            return (<Swatches hexIn={chrome.hex()} triggerSort={this.triggerSort} typ="hks" v-on:changed={this.textIn}/>);
20        case "name":
21            return (<Swatches hexIn={chrome.hex()} triggerSort={this.triggerSort} typ="name" v-on:changed={this.textIn}/>);
22        case "pantone":
23            return (<Swatches hexIn={chrome.hex()} triggerSort={this.triggerSort} typ="pantone" v-on:changed={this.textIn}/>);
24        default: return (<div>Hello</div>);
25    }
26 }
```

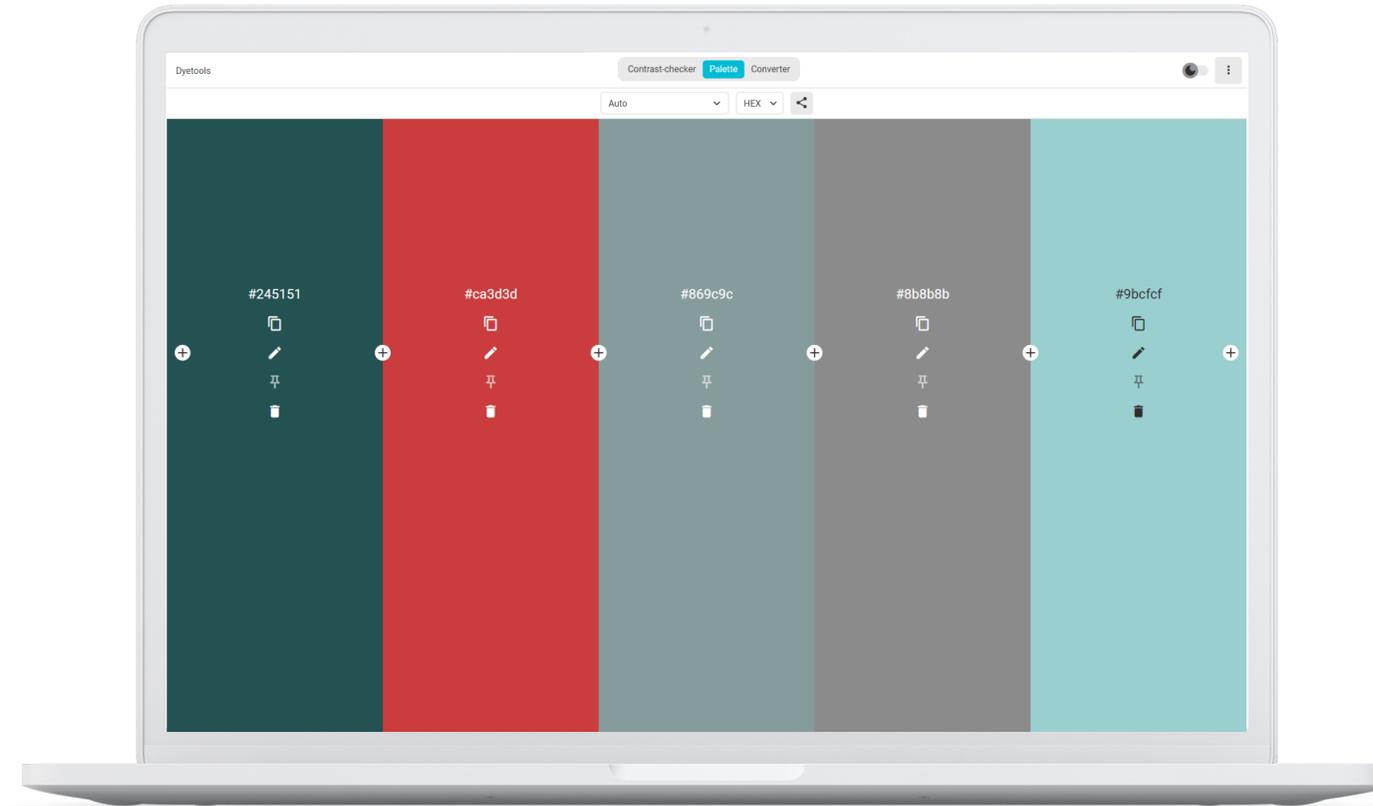
Da wir beide zuvor noch nie mit Vue / nuxt gearbeitet haben, waren gerade die ersten Wochen oftmals anstrengend. Trotzdem hat das Projekt viel Spaß gemacht und wir haben in den vergangenen Monaten viel dazugelernt



Mit dem Color Contrast Checker können zwei Farben auf ihren Kontrast überprüft werden. Die Ergebnisse sind nach WCAG-Standard eingestuft und bewertet und geben so einfache visuelle Rückmeldung über die Kompatibilität der ausgewählten Farben



Mit dem Farbconverter lässt sich eine ausgewählte Farbe in verschiedene Darstellungen konvertieren lassen. Über die „Lighten“- , „Darken“- , „Saturate“- und „Desaturate“-Slider lassen sich außerdem noch kleine Veränderungen an der Farbe vornehmen



Mit dem Farbpaletten-Generator lassen sich Farbpaletten mit bis zu 10 Farben auf Basis verschiedener Kontraste generieren. Einzelne Farben können bearbeitet, gelocked, entfernt und kopiert werden. Über die „+“-Buttons lassen sich an der ausgewählten Stelle Farbfelder einfügen