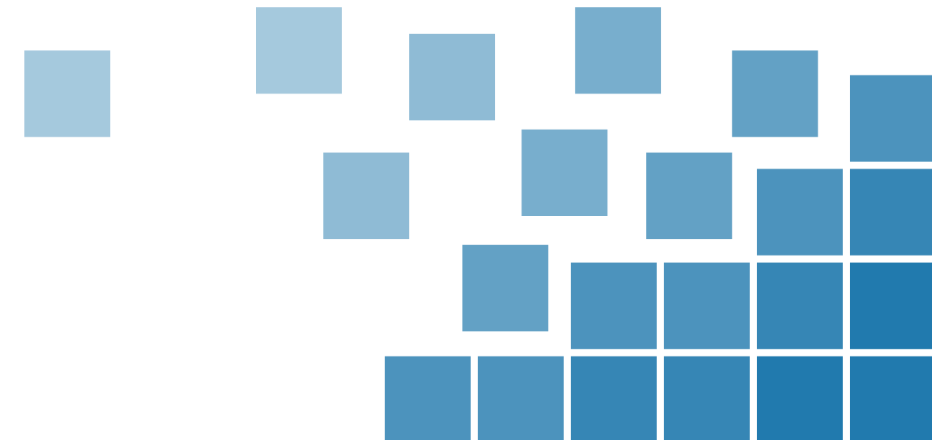


colorlendar

colorlendar ist eine Webapp, die hilft das aktuelle Wetter anschaulich und leichtverständlich mit den Vorjahren zu vergleichen. Dabei wollen wir auf die Veränderungen des Klimas aufmerksam machen und das vergangene Wetter in Bremen für mehr Menschen zugänglich und interessant machen.

Der colorlendar gibt jedem Monat, bzw. Tag anhand der Durchschnittstemperatur eine Farbe, die vorher anhand einer Skala ausgewählt wird. Blau ist kalt, Rot ist warm. Diese können dann in verschiedene Produkte eingearbeitet werden und zum Beispiel als Geschenke dienen. Dadurch wird der alltägliche Gebrauch des colorlendars gesichert!



Unsere Zielsetzung

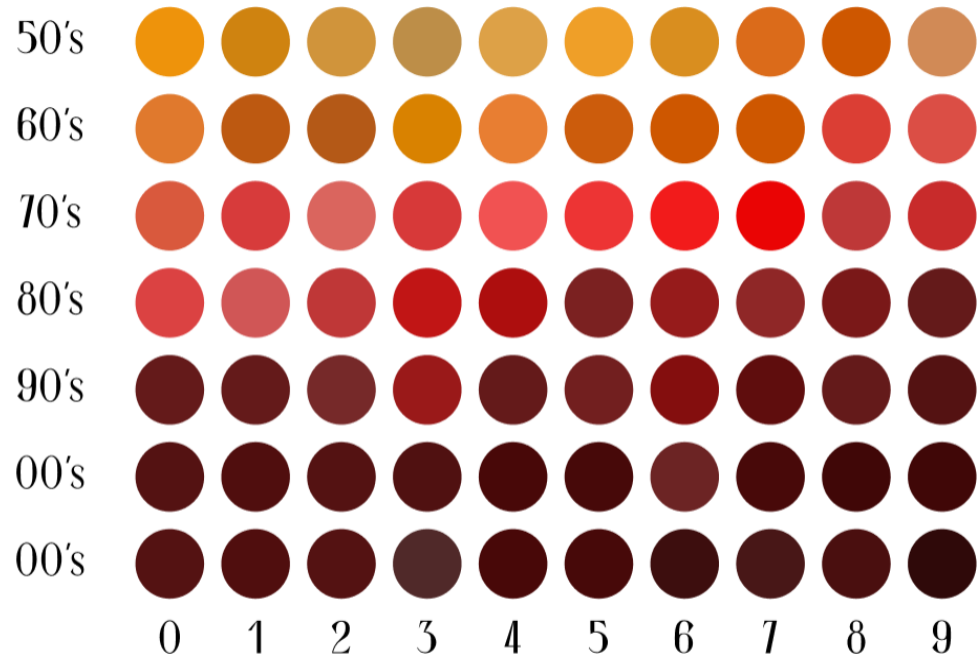
Bevor wir mit der Entwicklung begonnen, haben wir uns mit einem Problem auseinander gesetzt, welches wir bearbeiten wollen: Klimawandel.

- Wie kann man das so einfach und verständlich wie möglich machen? Minimalistisch und hell?
- Wie können wir es so gestalten, dass wir Informationen darüber teilen möchten?
- Wie kommerzialisieren wir die Datenvisualisierung?
- Wie macht man das Wetter anschaulich und nicht langweilig?



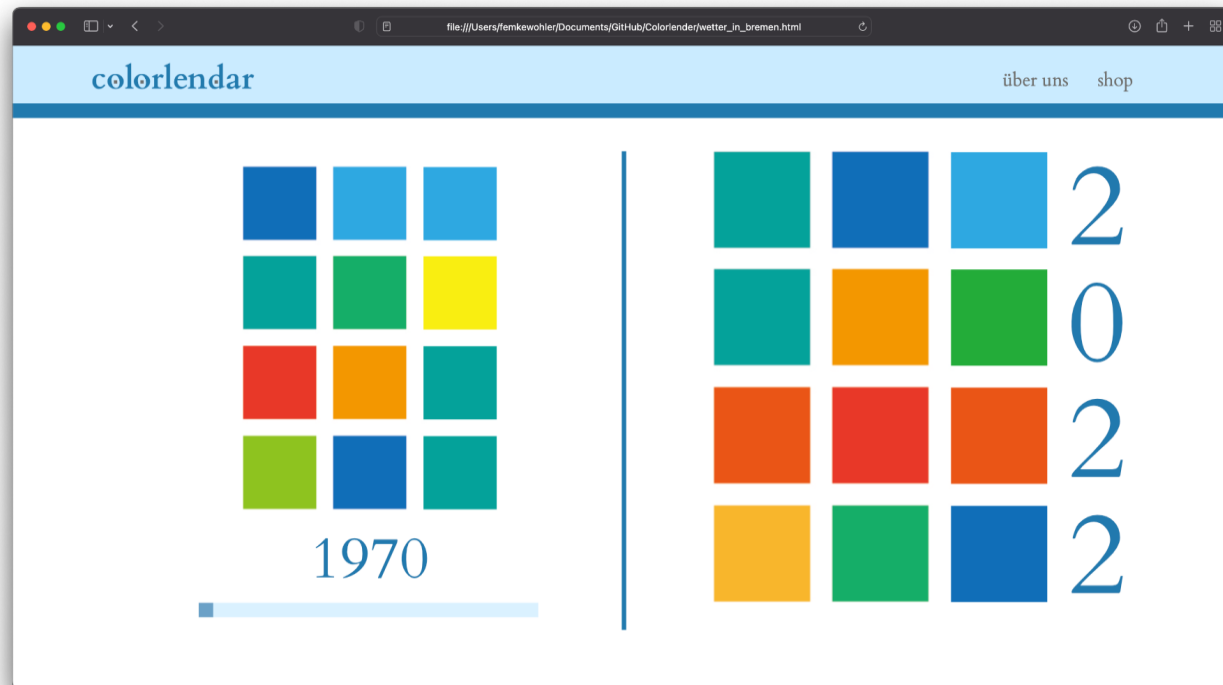
Das Design

- Erste Ideen. Warum haben wir Quadrate gewählt?



Das Design

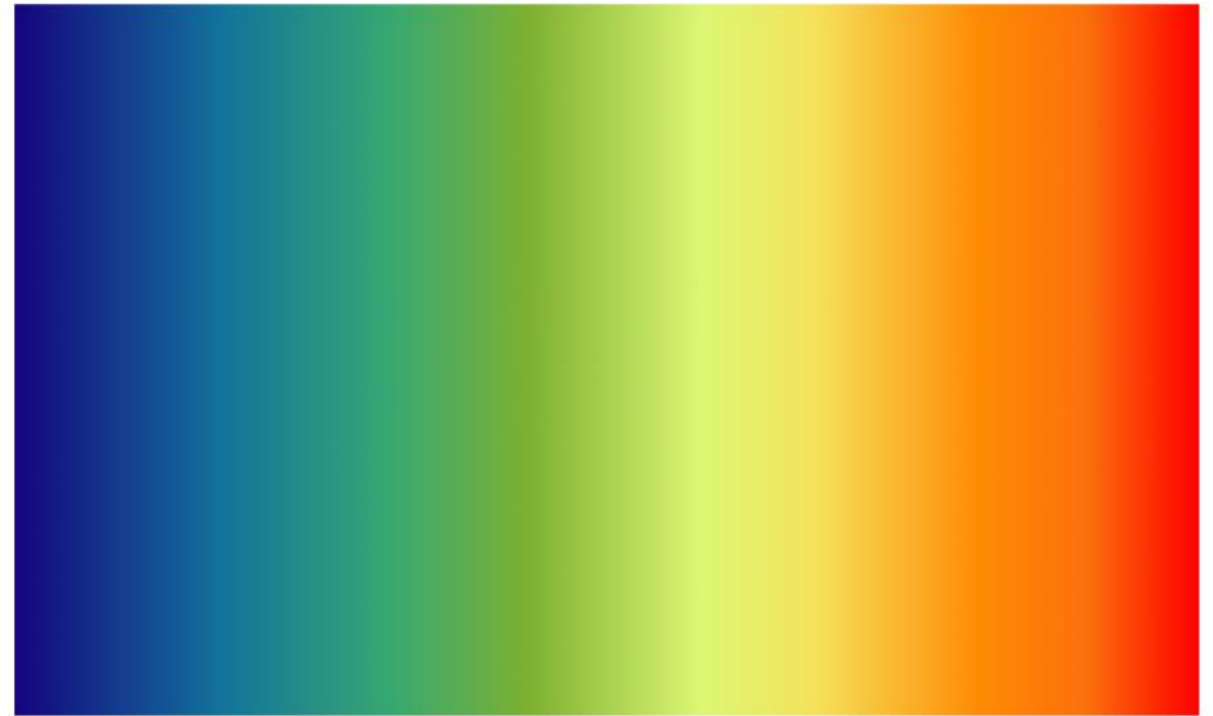
- Zur endgültigen Entscheidung gekommen



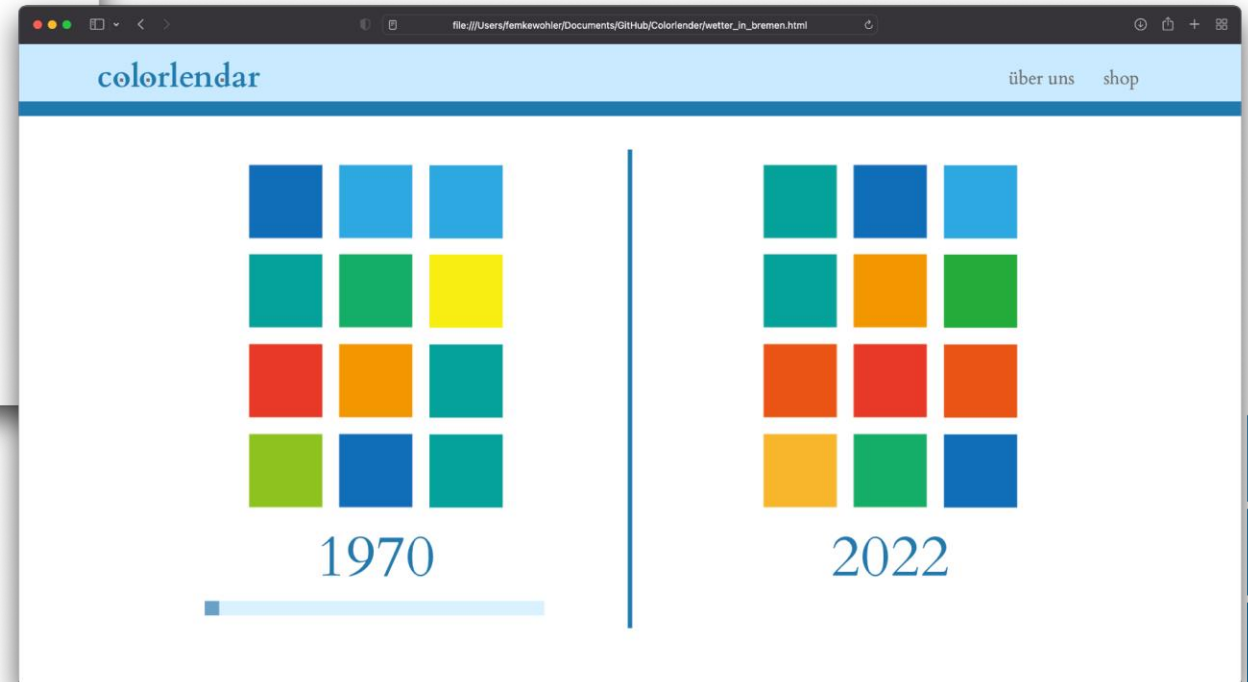
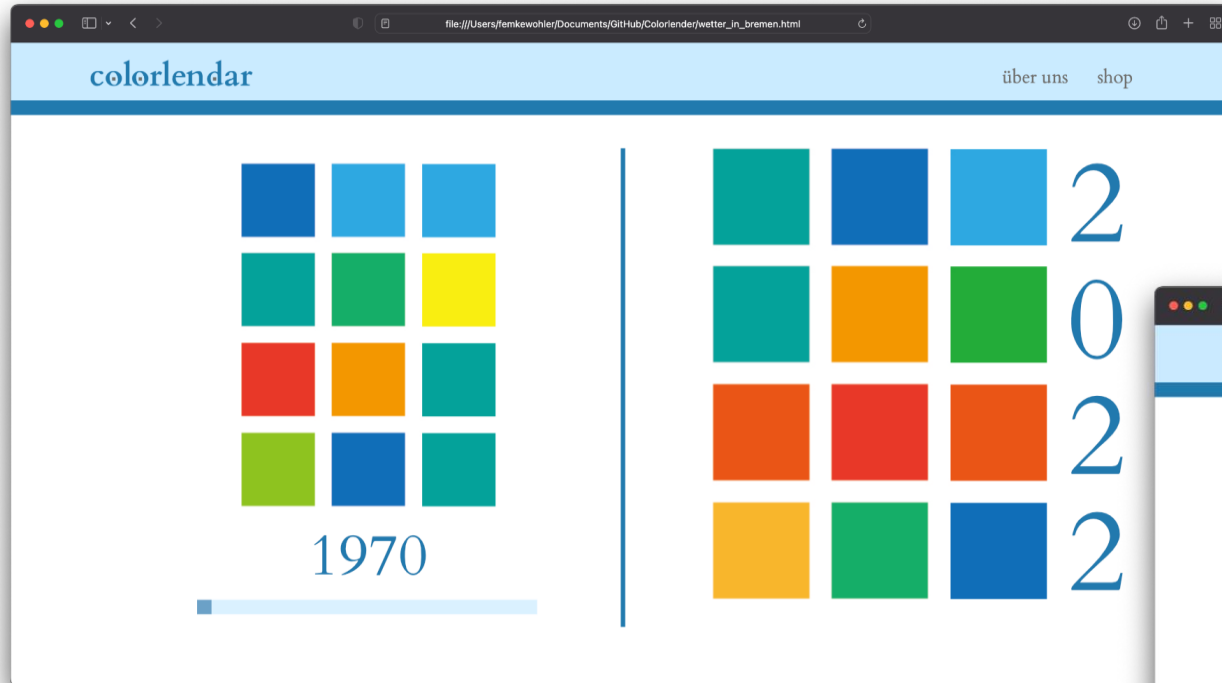
Die Farben

- Unsere gewählten Farben sind leicht verständlich, da wir uns an Farbassoziationen des Wetterberichts orientieren
- Wir wollen Informationen auf einen Blick klar machen

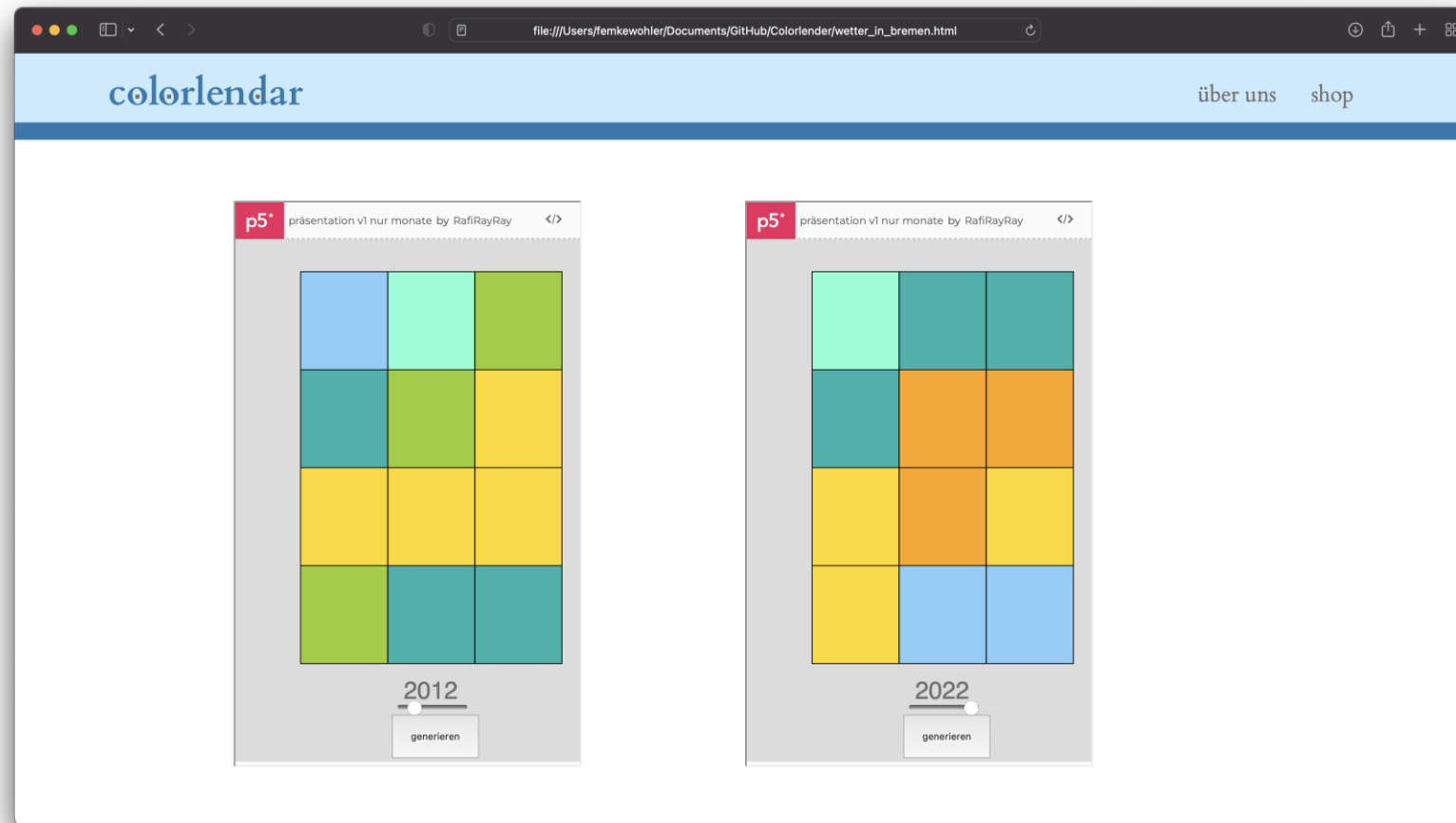
-20 -10 -5 0 +5 +10 +20 +30



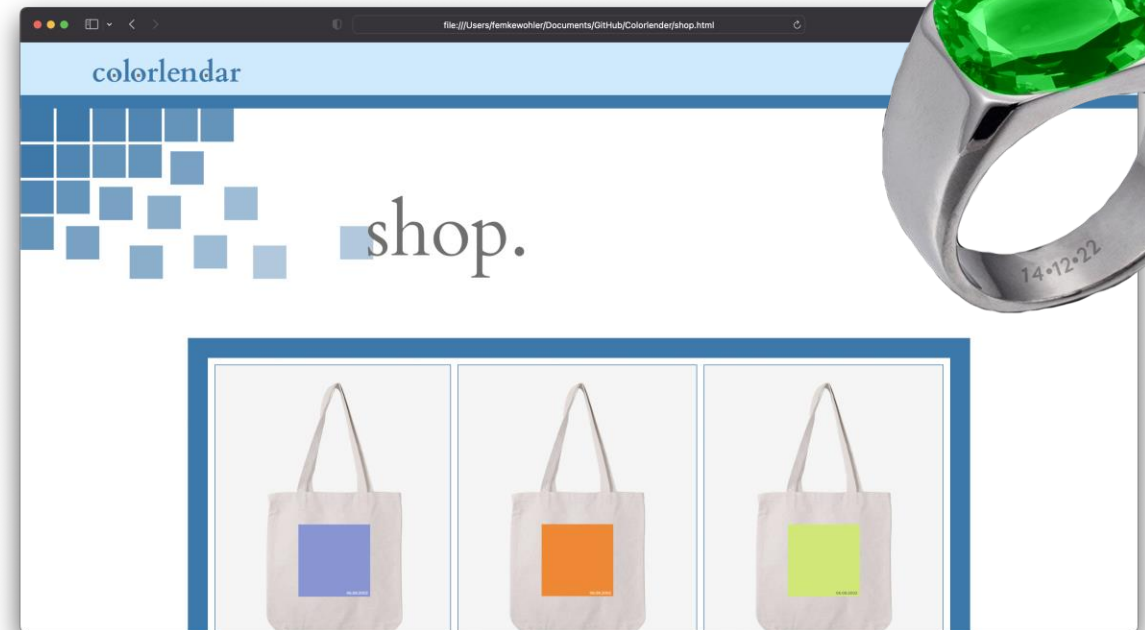
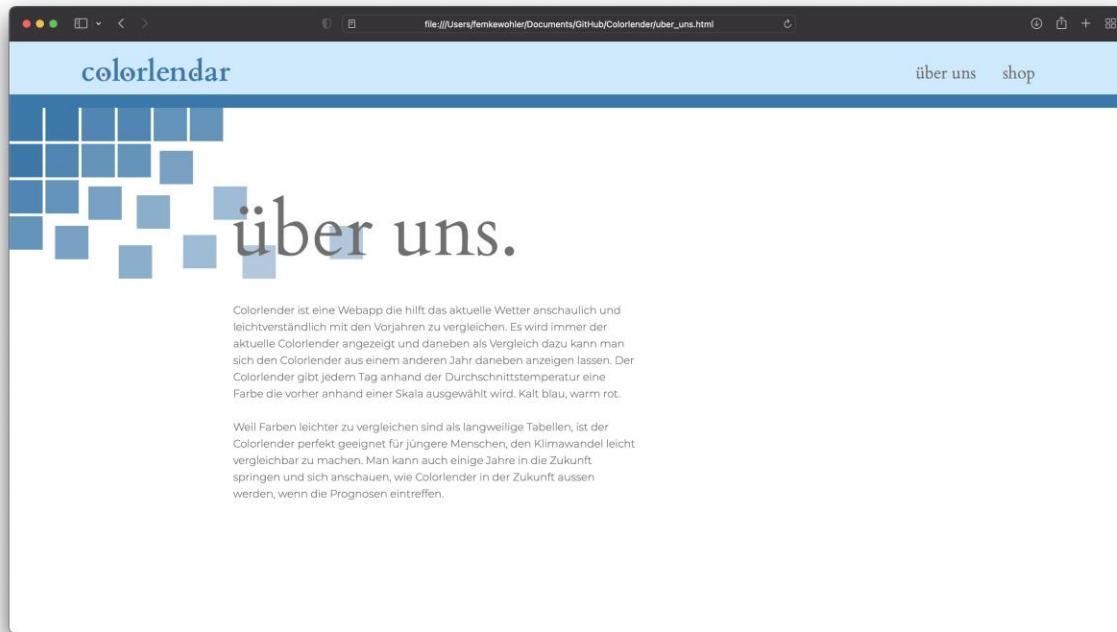
Graphical User Interface



Graphical User Interface

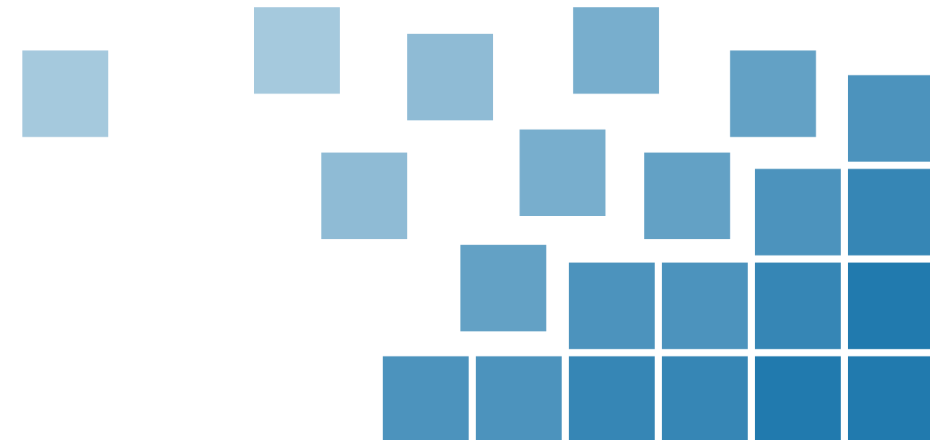


Graphical User Interface



p5

- Warum p5.js?
 - Javascript und sehr einfach einzubetten
- Daten kommen von brightsky API
 - Daten müssen für jeden Tag / Monat abgerufen und zwischengespeichert werden



Wichtige Stellen im Code

API laden

```
let url1 = api + j + "-" + "01" + "-" + t;  
januar = loadJSON(url1, m1);  
  
let url2 = api + j + "-" + "02" + "-" + t;  
februar = loadJSON(url2, m2);  
  
let url3 = api + j + "-" + "03" + "-" + t;  
marz = loadJSON(url3, m3);  
  
let url4 = api + j + "-" + "04" + "-" + t;  
april = loadJSON(url4, m4);  
  
let url5 = api + j + "-" + "05" + "-" + t;  
mai = loadJSON(url5, m5);  
  
let url6 = api + j + "-" + "06" + "-" + t;  
juni = loadJSON(url6, m6);  
  
let url7 = api + j + "-" + "07" + "-" + t;  
juli = loadJSON(url7, m7);  
  
let url8 = api + j + "-" + "08" + "-" + t;  
august = loadJSON(url8, m8);  
  
let url9 = api + j + "-" + "09" + "-" + t;  
september = loadJSON(url9, m9);  
  
let url10 = api + j + "-" + "10" + "-" + t;  
oktober = loadJSON(url10, m10);  
  
let url11 = api + j + "-" + "11" + "-" + t;  
november = loadJSON(url11, m11);  
  
let url12 = api + j + "-" + "12" + "-" + t;  
dezember = loadJSON(url12, m12);
```

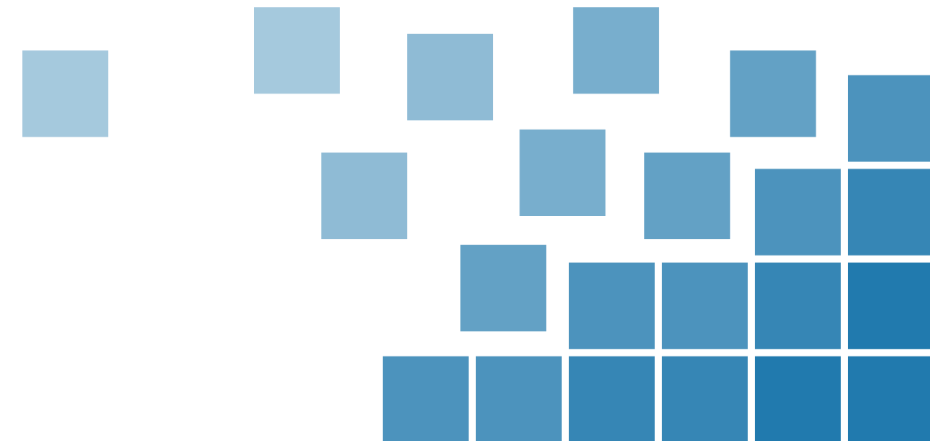
Daten zwischenspeichern

```
function m1(januar) {  
  t1 = januar.weather[11].temperature;  
  loaded1 = true;  
}  
function m2(februar) {  
  t2 = februar.weather[11].temperature;  
  loaded2 = true;  
}  
function m3(marz) {  
  t3 = marz.weather[11].temperature;  
  loaded3 = true;  
}  
function m4(april) {  
  t4 = april.weather[11].temperature;  
  loaded4 = true;  
}  
function m5(mai) {  
  t5 = mai.weather[11].temperature;  
  loaded5 = true;  
}  
function m6(juni) {  
  t6 = juni.weather[11].temperature;  
  loaded6 = true;  
}  
function m7(juli) {  
  t7 = juli.weather[11].temperature;  
  loaded7 = true;  
}  
function m8(august) {  
  t8 = august.weather[11].temperature;  
  loaded8 = true;  
}
```

Temperatur -> Farbe

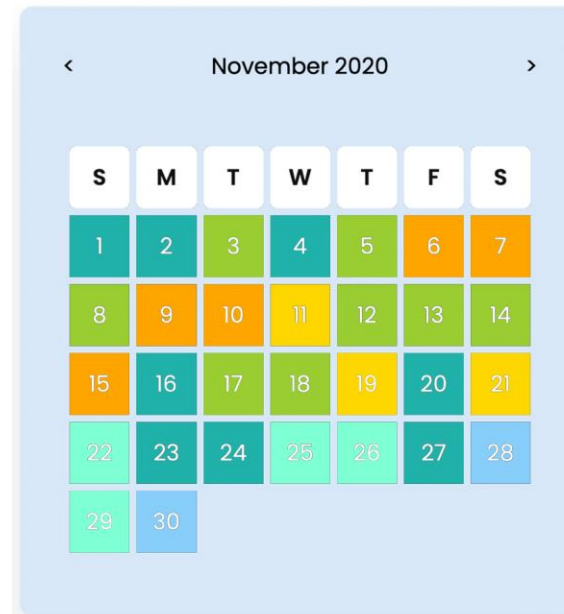
```
if (loaded1) {  
  if (t1 < -10) {  
    fill(072, 061, 139);  
  } else if (t1 > -10 && t1 < -5) {  
    fill(0, 0, 205);  
  } else if (t1 > -5 && t1 < 0) {  
    fill(135, 206, 250);  
  } else if (t1 > 0 && t1 < 5) {  
    fill(127, 255, 212);  
  } else if (t1 > 5 && t1 < 10) {  
    fill(032, 178, 170);  
  } else if (t1 > 10 && t1 < 15) {  
    fill(154, 205, 050);  
  } else if (t1 > 15 && t1 < 20) {  
    fill(255, 215, 0);  
  } else if (t1 > 20 && t1 < 25) {  
    fill(255, 165, 0);  
  } else if (t1 > 30 && t1 < 35) {  
    fill(255, 069, 0);  
  } else if (t1 > 35) {  
    fill(255, 0, 0);  
  }  
}  
  
rect(rectX, rectY, rectWidth / 3, rectHeight / 4);
```

Live Demo



Die nächsten Schritte

- Mehr Daten
 - Auch Daten vor 2010 und Prognosen für die Zukunft einfügen
- Visuelle Verbesserungen
 - Interessante Vergleiche
 - Monat -> Tag



Referenzen

Wetterdatenbank: brightsky.dev

Bilder für den Shop: pexels.com

