



15 years ago...





7 years ago...

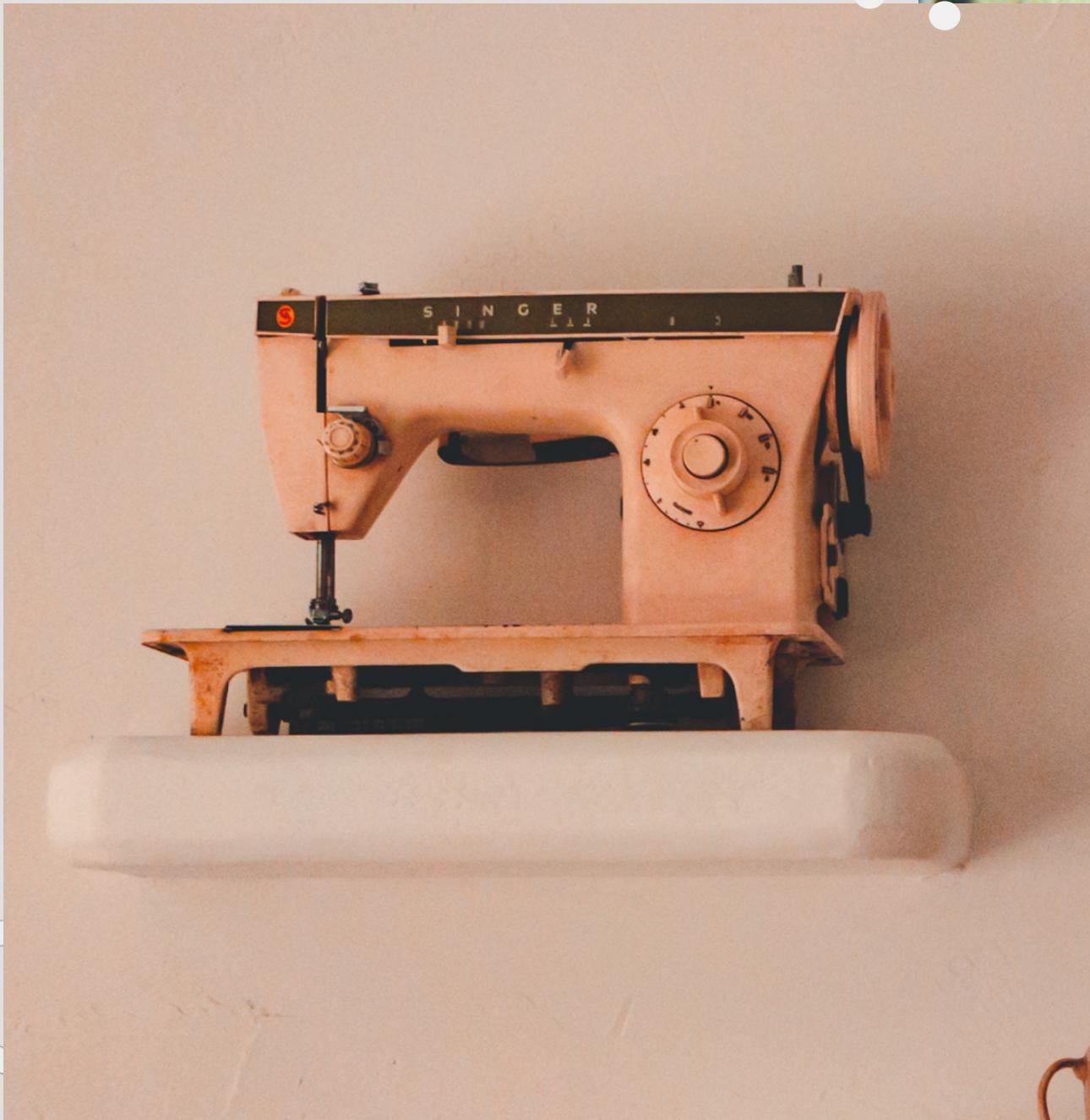
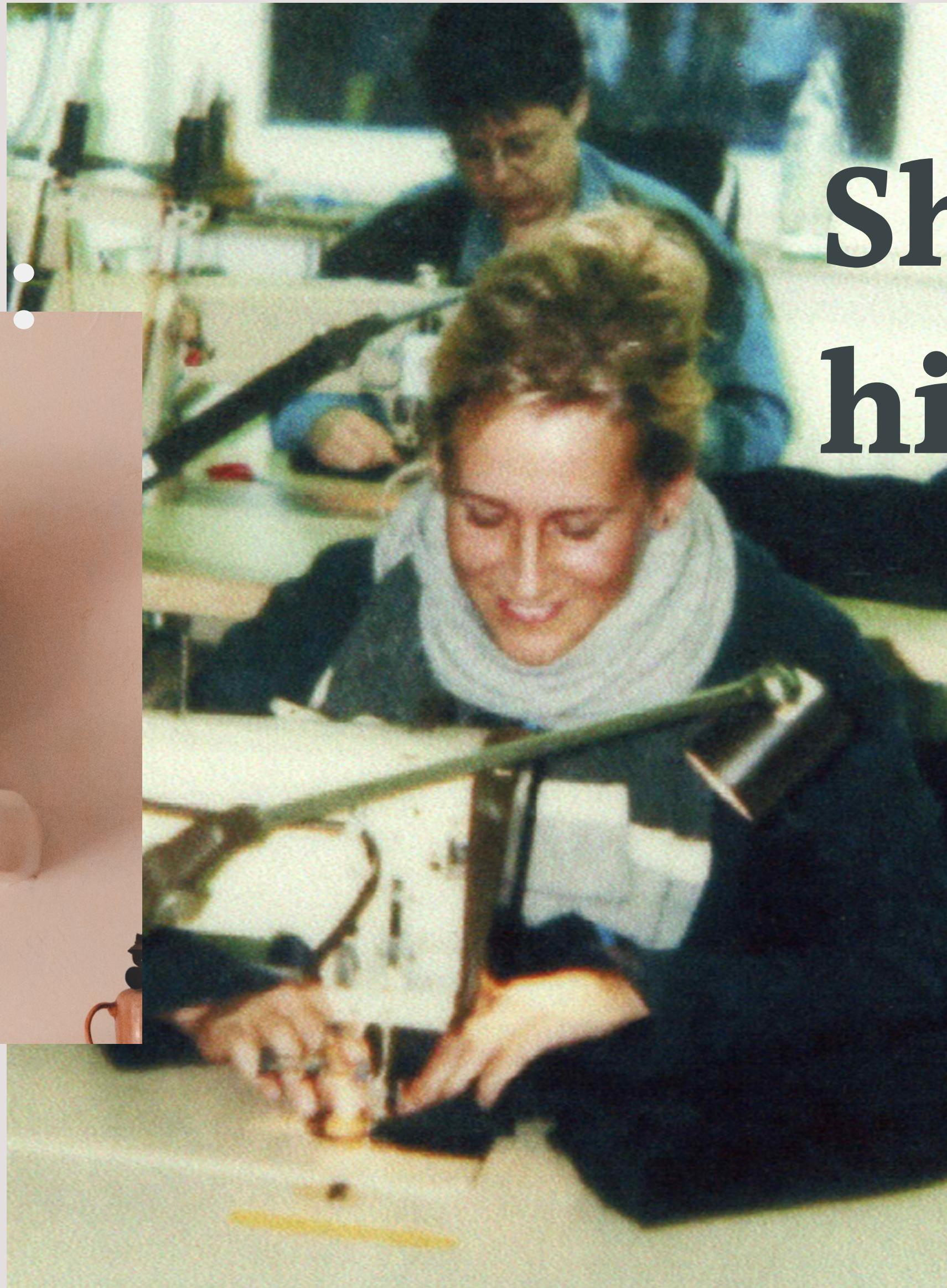
2012







She had a long
history...

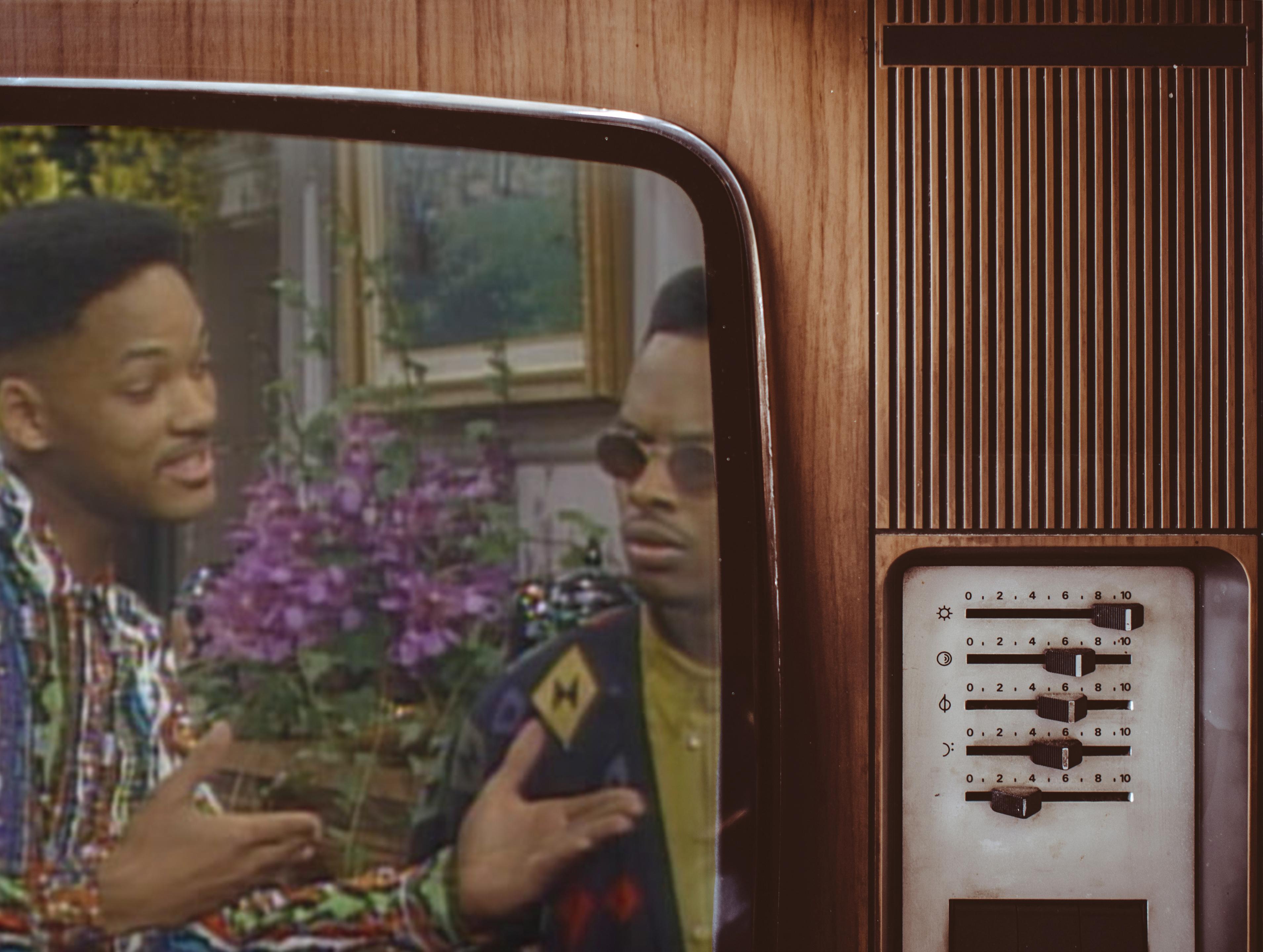


The grandma



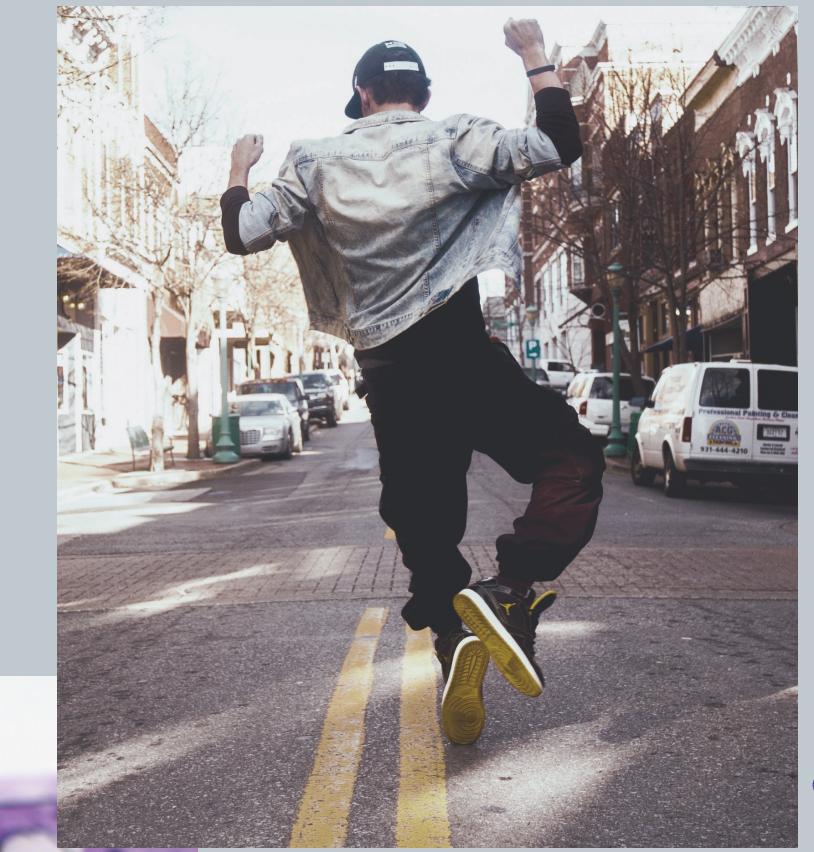
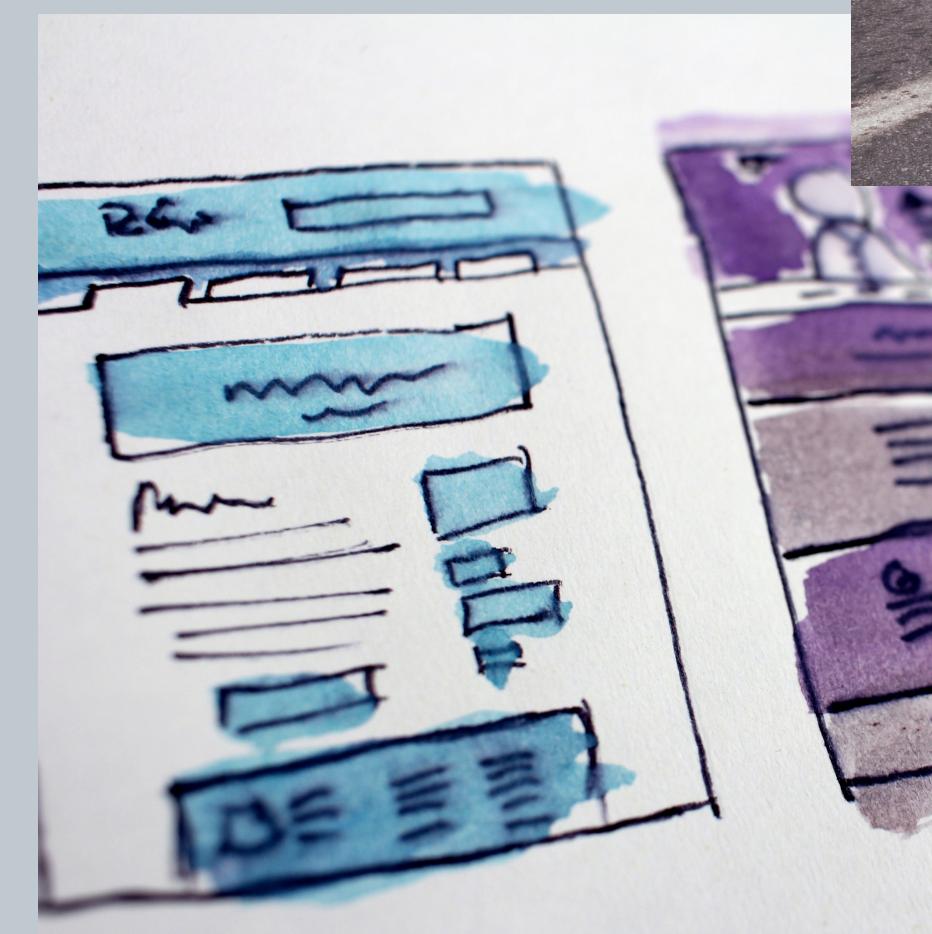


From



To

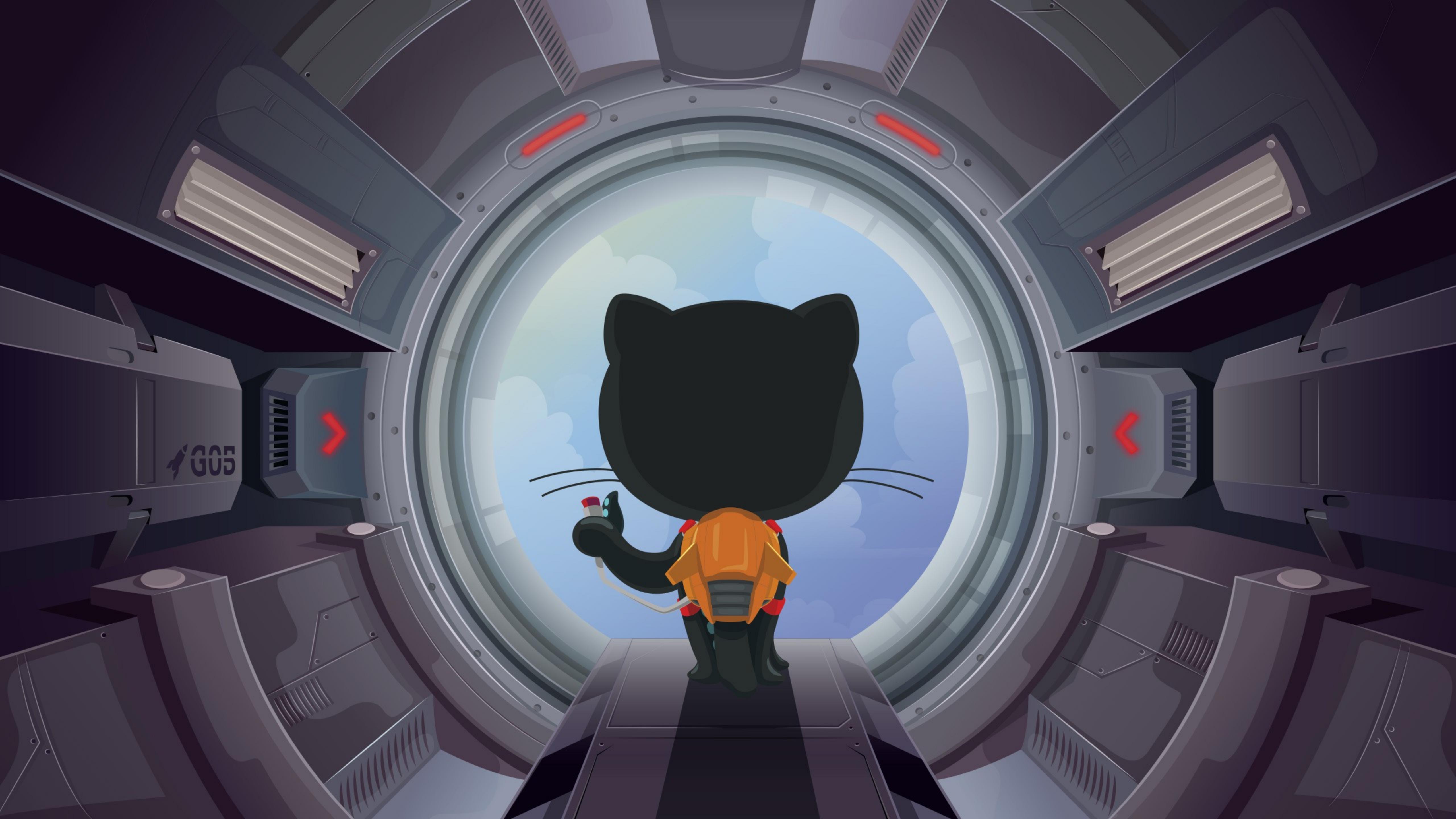
They founded their label and...



The Man







Today

Mauricio Palma

Co-founder and all kind of things @WDLK

Product Engineer @sinnerschrader

everywhere else @palmaswell





Why?



Back in The Day





Way
harder!

opportunity



Nature of the web





We use our
our abilities

The background of the slide features a photograph of a sunset or sunrise. The sky is a gradient from deep blue at the top to a warm orange and yellow near the horizon. Silhouettes of mountains are visible against the bright sky.

You can't read
this sentence

A11y automation

gAnead

1. iVsual dsibailtiise
2. oColr cnortsat
3. Rletaive ulimannce
4. oClro ilsts
5. uQciskrot
6. iBanry esrahc
7. aHsh atble
8. Dmeo

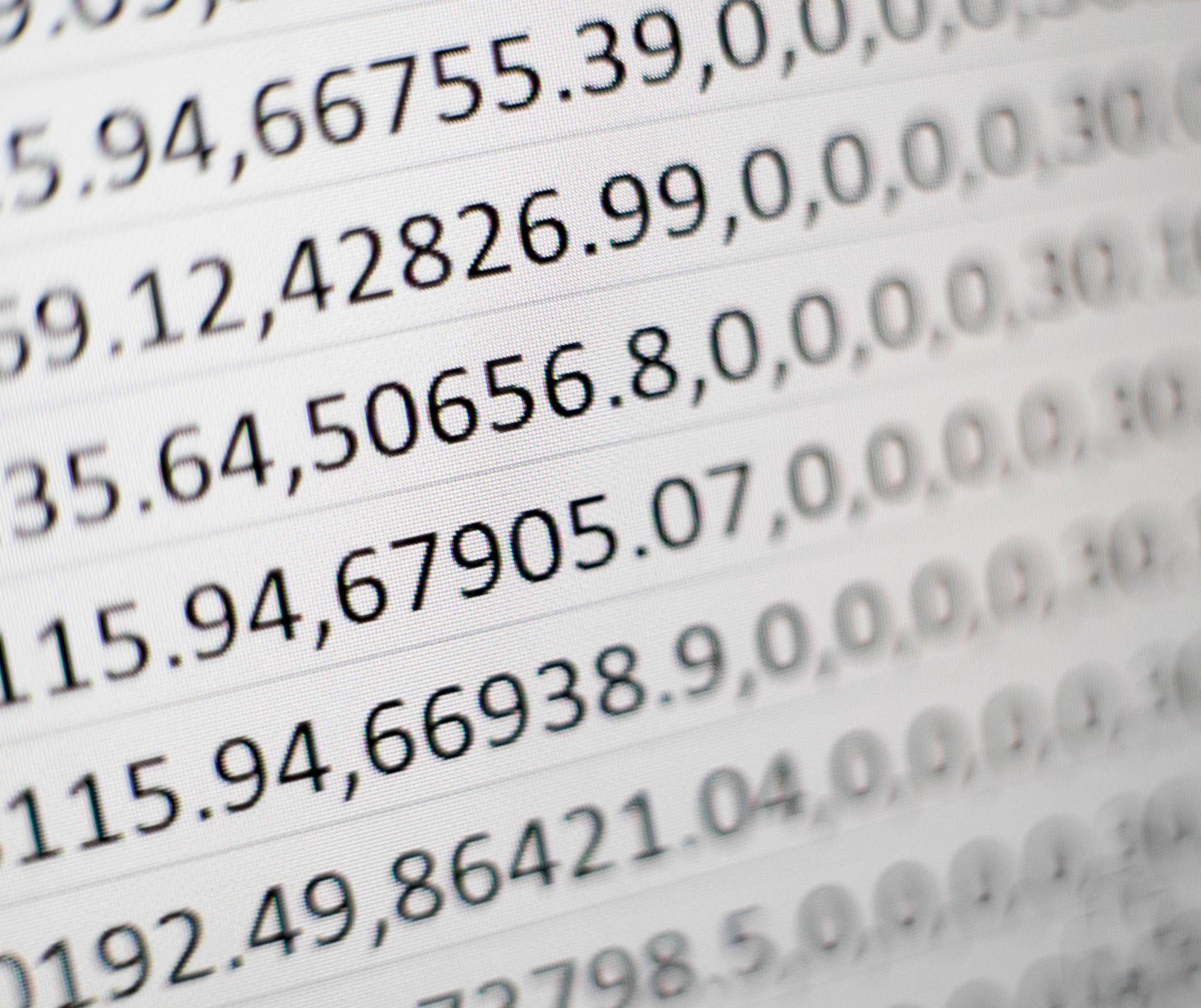
That was
awkward! Right?



Literally
hurts!

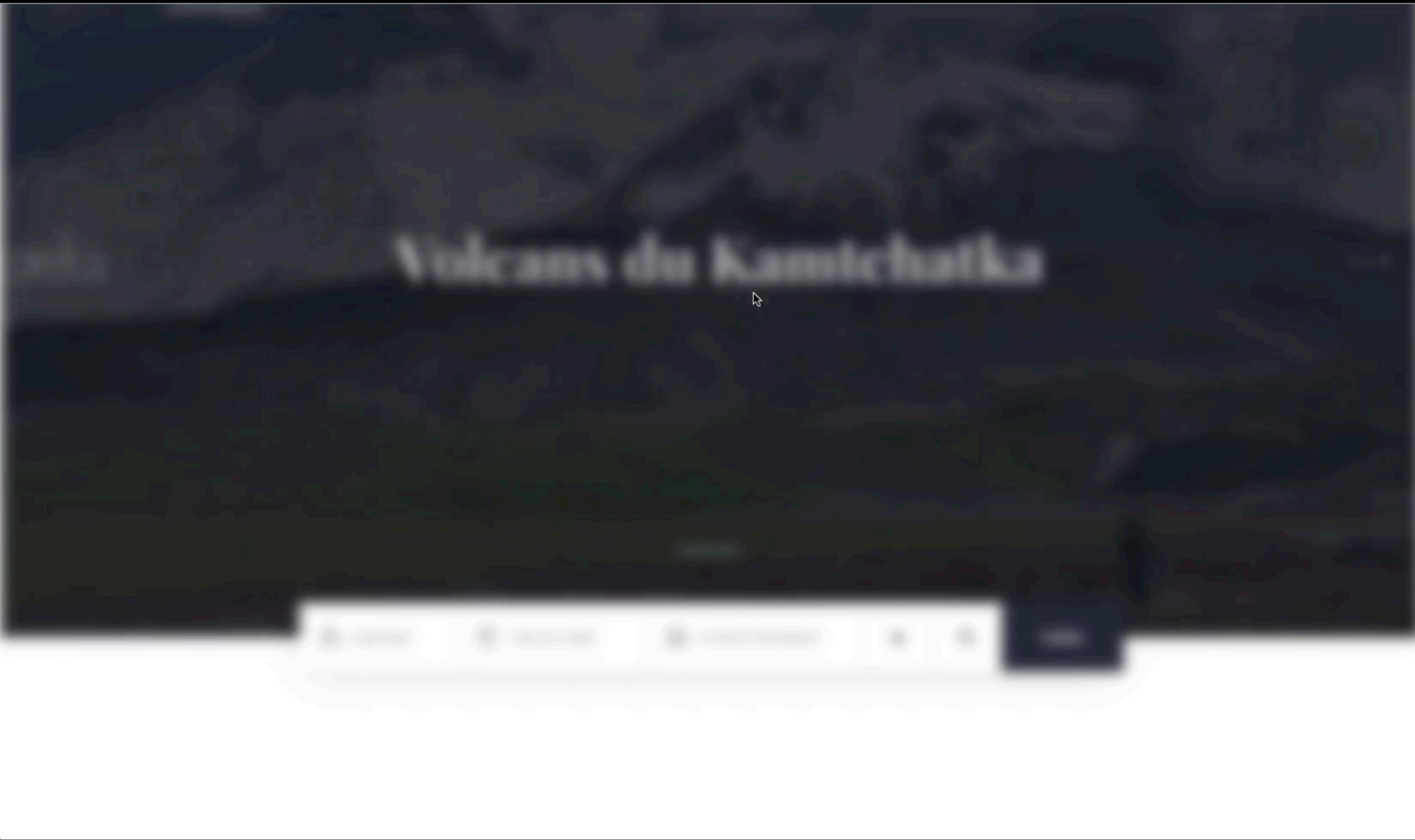


217 million



How to
fix the
numbers?

Blurry Vision



Vulkane der Kamtschatka

→

Key Takeaways

- ⚡ Publish all information and content from your site in your HTML
- ⚡ Use semantic HTML and the appropriate ARIA landmarks
- ⚡ Write screen reader friendly markup
- ⚡ Provide keyboard navigation

Partial vision loss

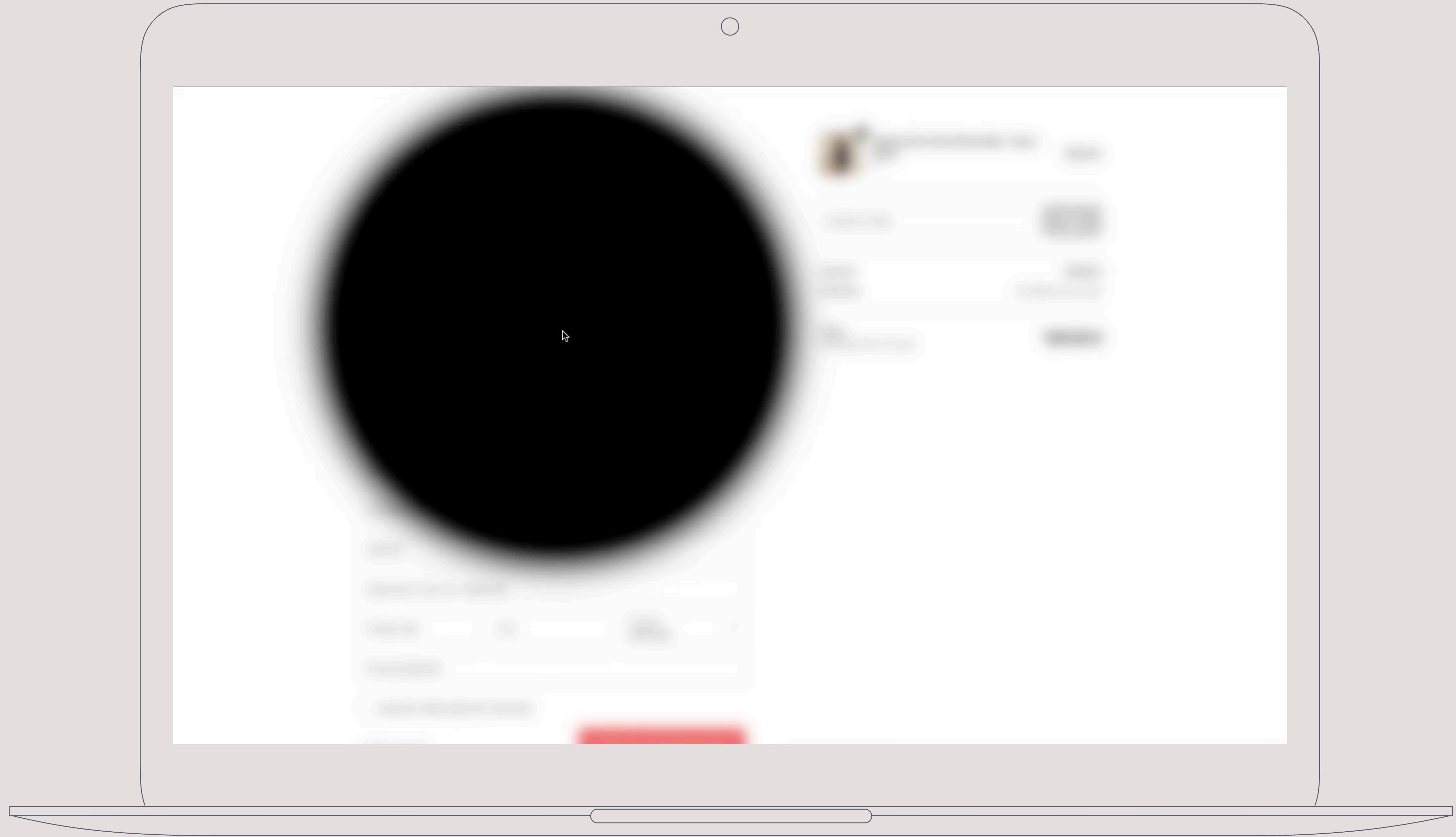
Makin

One of
form

Key Takeaways

- ⚡ Besides the mentioned semantic HTML and screen reader friendly markup
- ⚡ You should follow a linear logical layout, that supports at least 200% magnification

Central vision loss



Key Takeaways

- ⚡ Always put form elements in context. Meaning putting buttons, inputs, and notifications together in a logical manner
- ⚡ For action elements in your interface use a good combination of colors, shapes, and text
- ⚡ Do not rely on colors to convey meaning

Dyslexia

rGaet rpoudcts **hcnaeg**
veretyihng. Bhevaiuor.
uBisenss. aMkrtes.

hTta's hwy we ublid rtasnofmrtainola rpdouts. Prdocust, adn es-
rcies, taht are ont noyl rgeat, ubt laso obnd iwth htier suesr, er-
shape uasge adn rtnasfrom bsuienss eraltiy. rPdouts that edilevr
uniue vlaeu rpoopsitoins opreed by well cartfed osftware, ewll
htough uot UX and isnpirign edsigns.

**WE AHVE 7 ASCERD URELS
TO BIULDING A CRAFT**

Key Takeaways

- ⚡ Align your text to the left and keep a consistent layout
- ⚡ Keep content short, clear and simple
- ⚡ Consider producing materials in other formats (for example audio or video)
- ⚡ Let the users change the contrast between background and text

Colorblindness

Beach pictures

21,642 free beach pictures



white hammock tied to coconut trees

Trees Solitude Palm



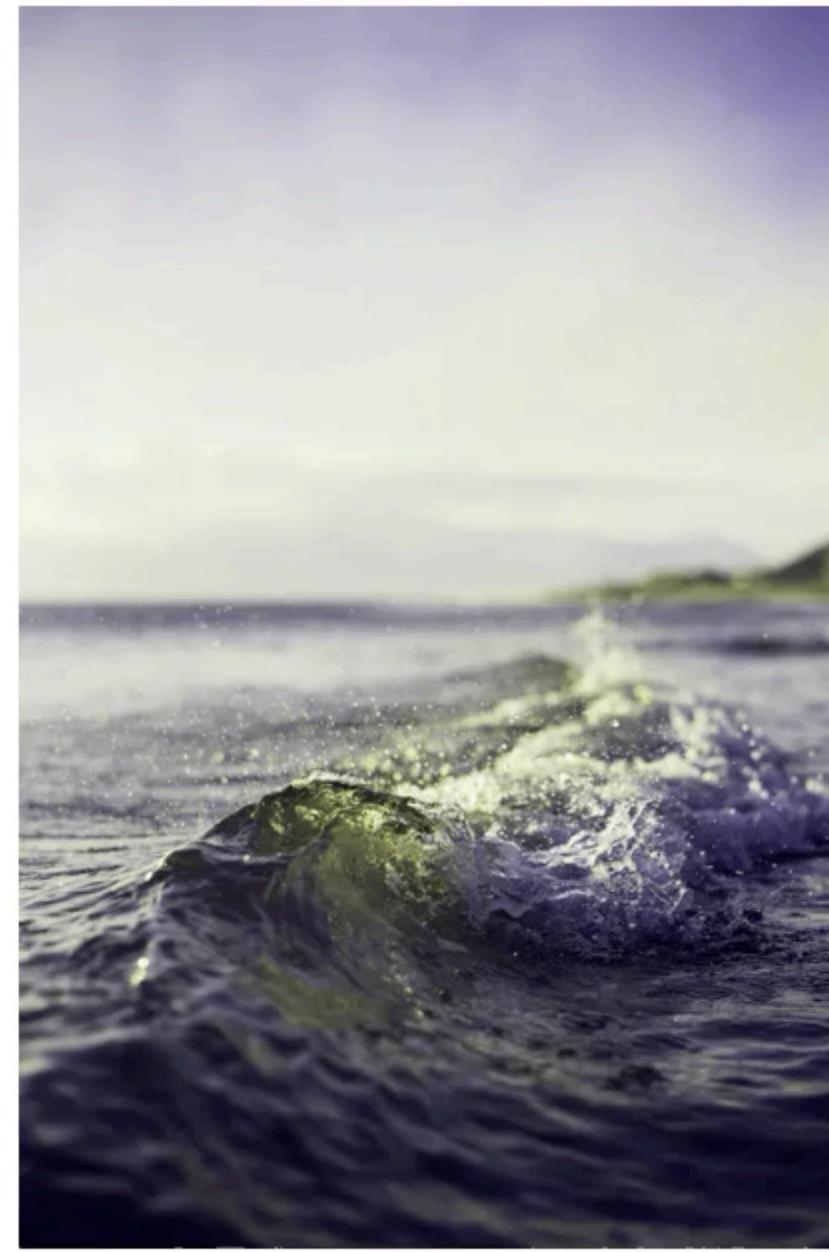
body of water during daytime

Cloud Ocean Sea



brown wooden walkway near beach during daytime

Sea Stairs Sand



ocean wave at beach

Wave Water Action

Beach pictures

21,642 free beach pictures



white hammock tied to coconut trees

Trees Solitude Palm



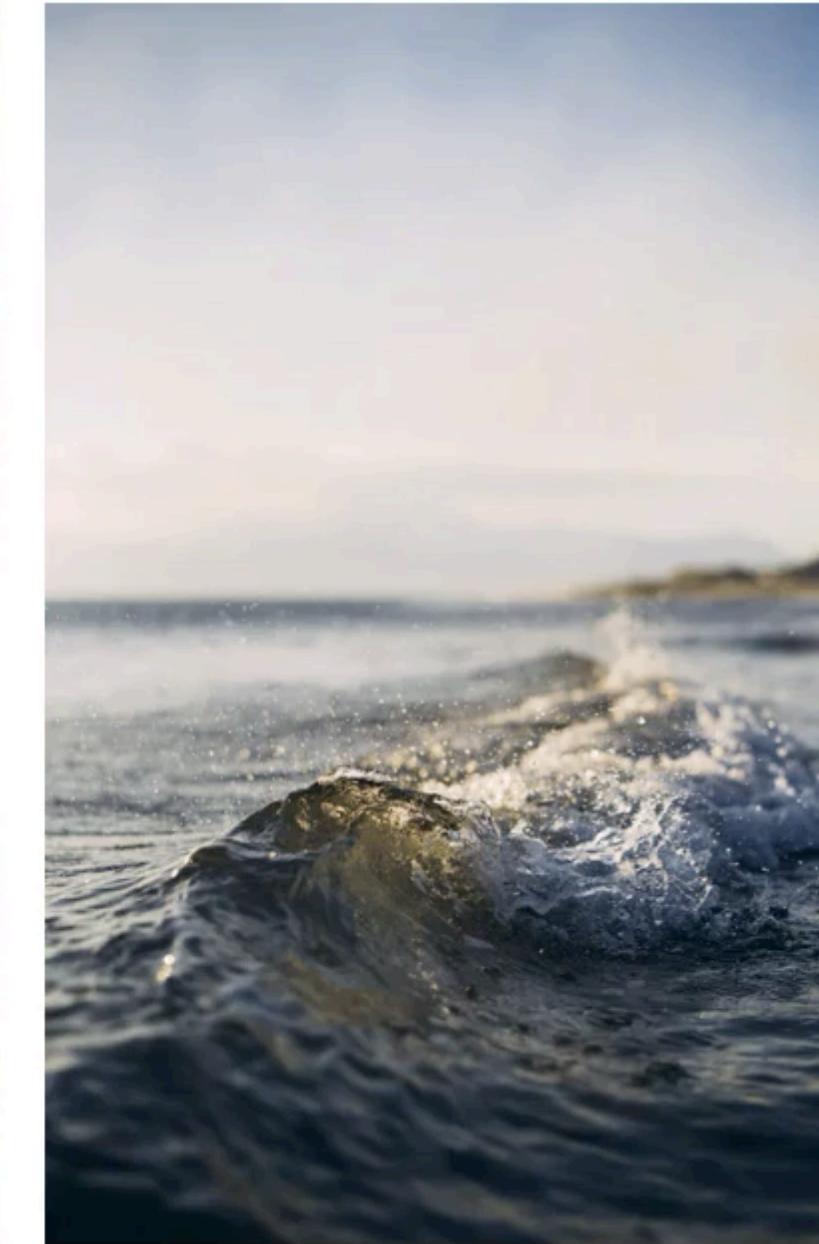
body of water during daytime

Cloud Ocean Sea



brown wooden walkway near beach during daytime

Sea Stairs Sand



ocean wave at beach

Wave Water Action

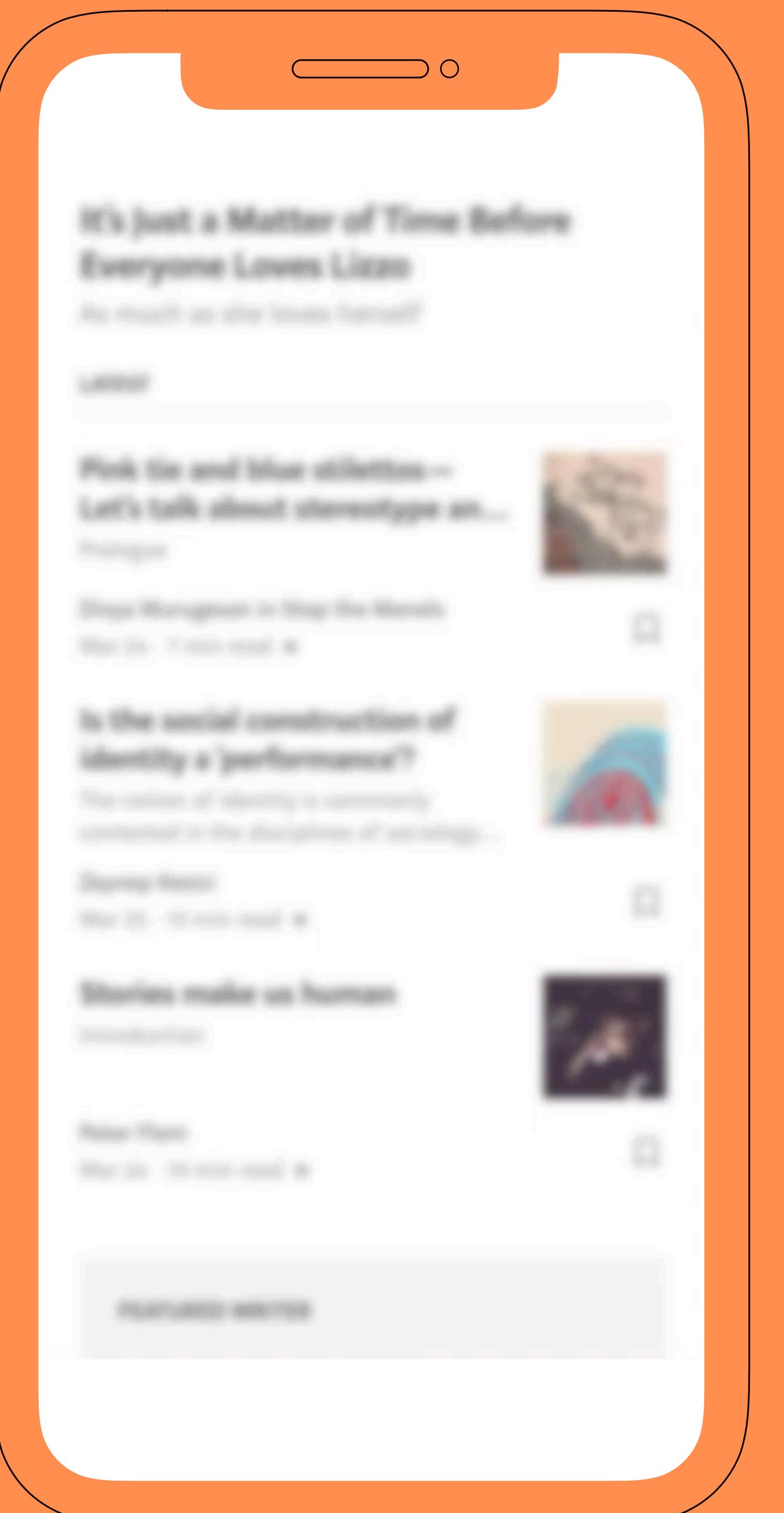
Key Takeaways

⚡ Make sure that colors are not your only method of conveying important information.

Sunlight

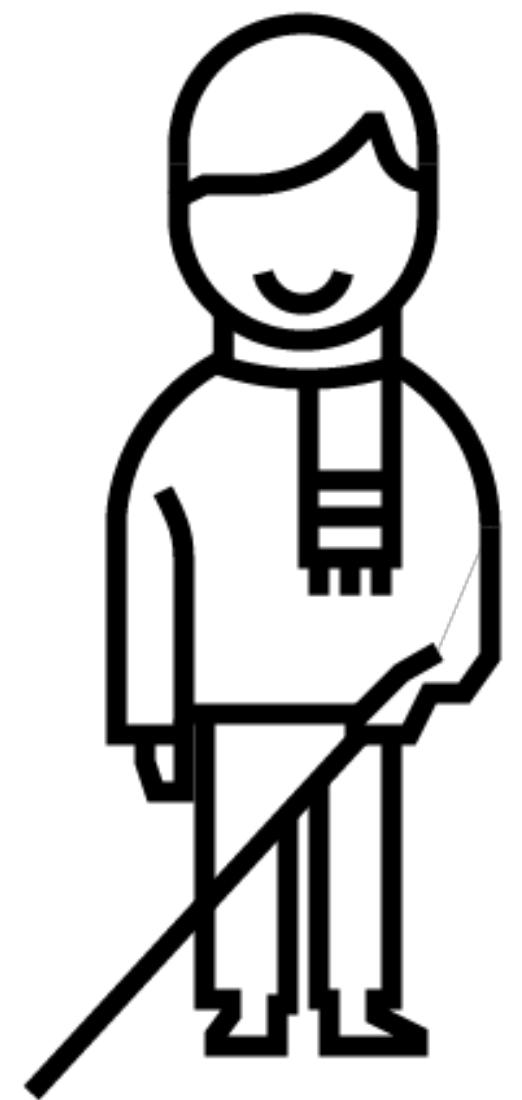
Bad lightning

Temporary disabilities



Persona spectrum

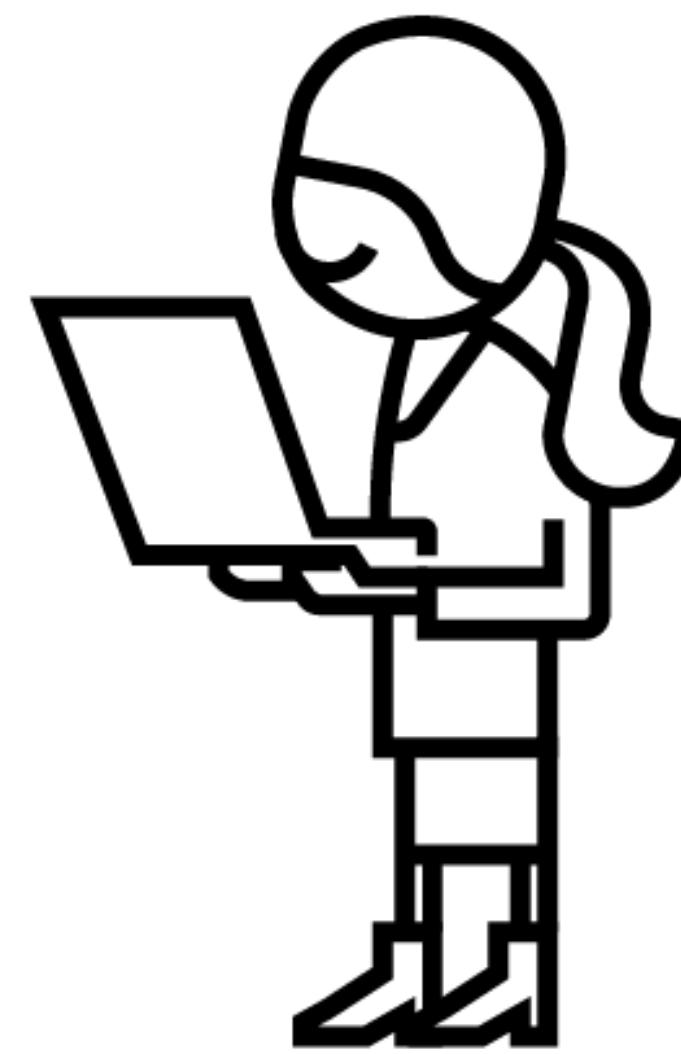
Permanent



Temporary

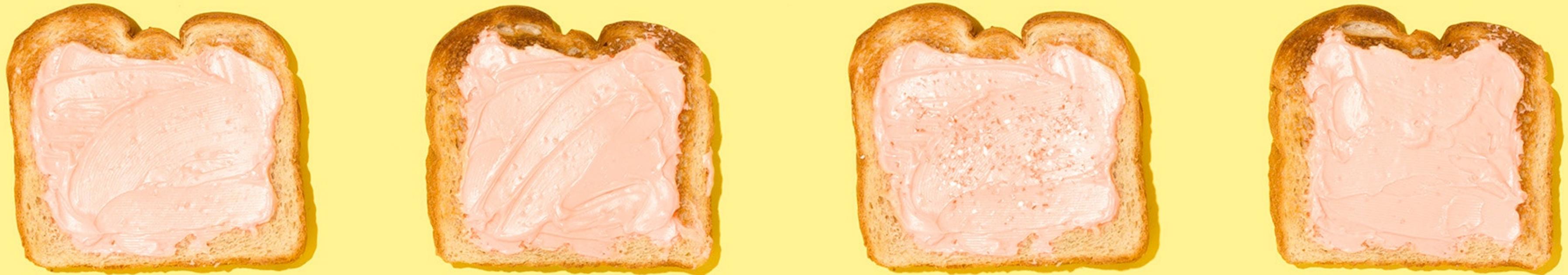


Situational

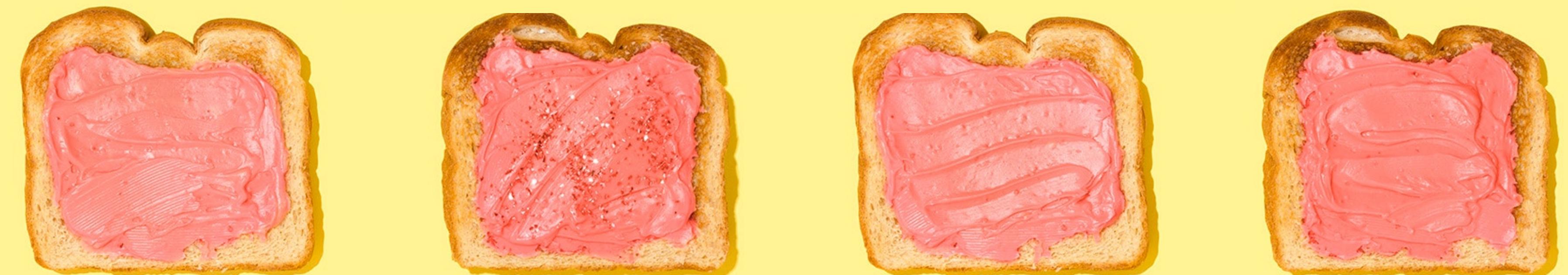


more people





A11y everywhere



You Code On Visit the Ship

```
import passport from 'passport';
import LocalStrategy from 'passport-local';
import { Strategy as JWTStrategy, ExtractJwt } from 'passport-jwt';
import request from 'superagent';

import User from './models/user.model';
import constants from '../config/constants';
import { createUser } from '../helpers/auth.helper';

/**
 * Local Strategy Auth
 */
const localOpts = { usernameField: 'username' };

const localLogin = new LocalStrategy(
  localOpts,
  async (username, password, done) => {
    try {
      const user = await User.query().where('username', username);

      if (user.length === 0) {
        const userData = {
          username,
          password,
        };
        const createdUser = await createUser(userData);
        return done(null, createdUser);
      }
      return done(null, user[0]);
    } catch (e) {
      return done(e, null);
    }
  }
);

/**
 * JWT Strategy Auth
 */
const jwtOpts = {
  // Telling Passport to check authorization headers for JWT
  jwtFromRequest: ExtractJwt.fromAuthHeaderWithScheme('JWT'),
  // Telling Passport where to find the secret
  secretOrKey: constants.JWT_SECRET,
};

const jwtLogin = new JWTStrategy(jwtOpts, async (payload, done) => {
  try {
    console.log(payload);
    const user = await User.query().where('user_uuid', payload.user_uuid).first();
    console.log(user.toJSON());

    if (user.length === 0 || !user) {
      return done(null, false);
    }

    return done(null, user[0]);
  } catch (e) {
    console.log(e);
    return done(e, false);
  }
});

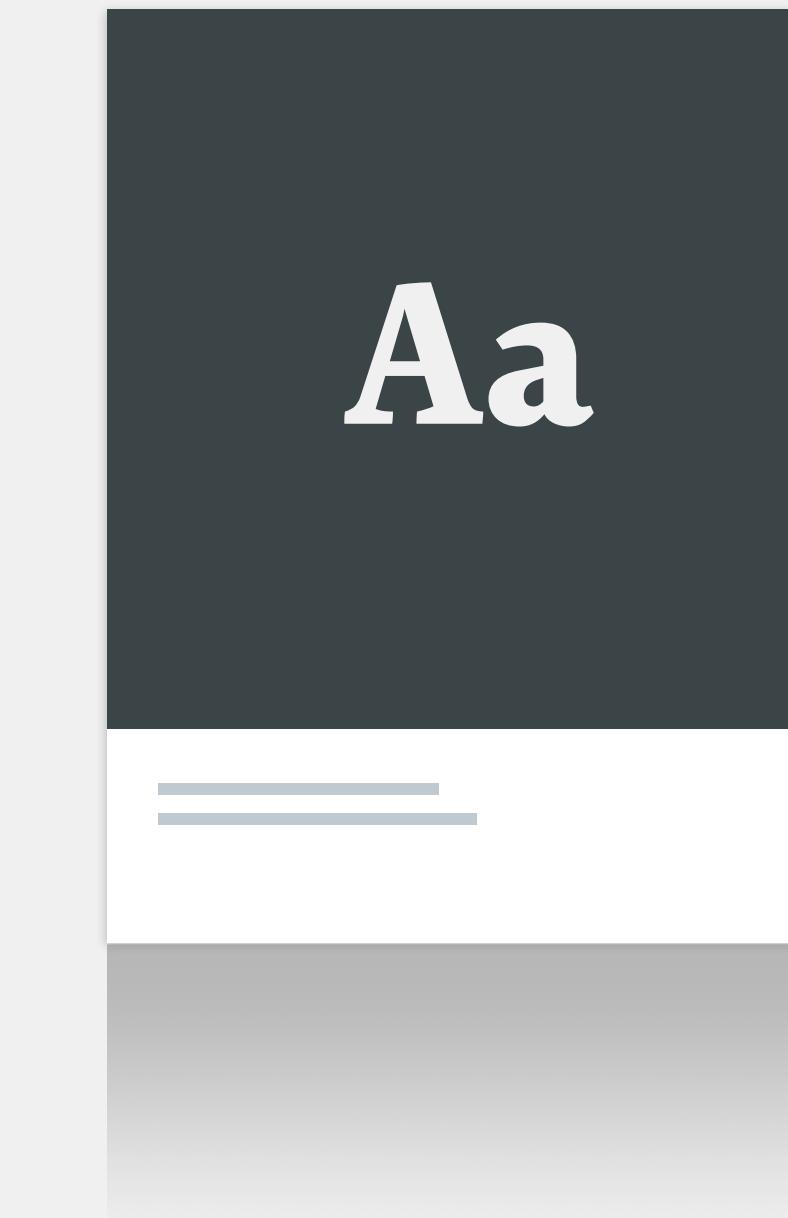
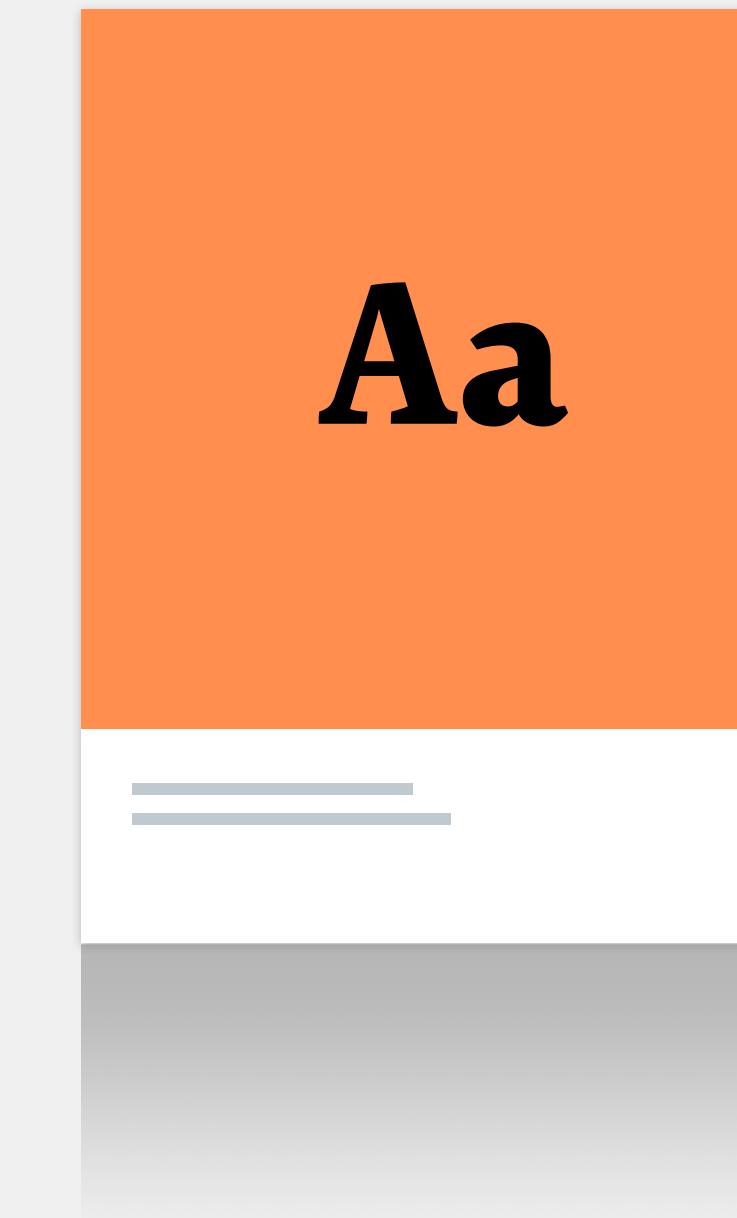
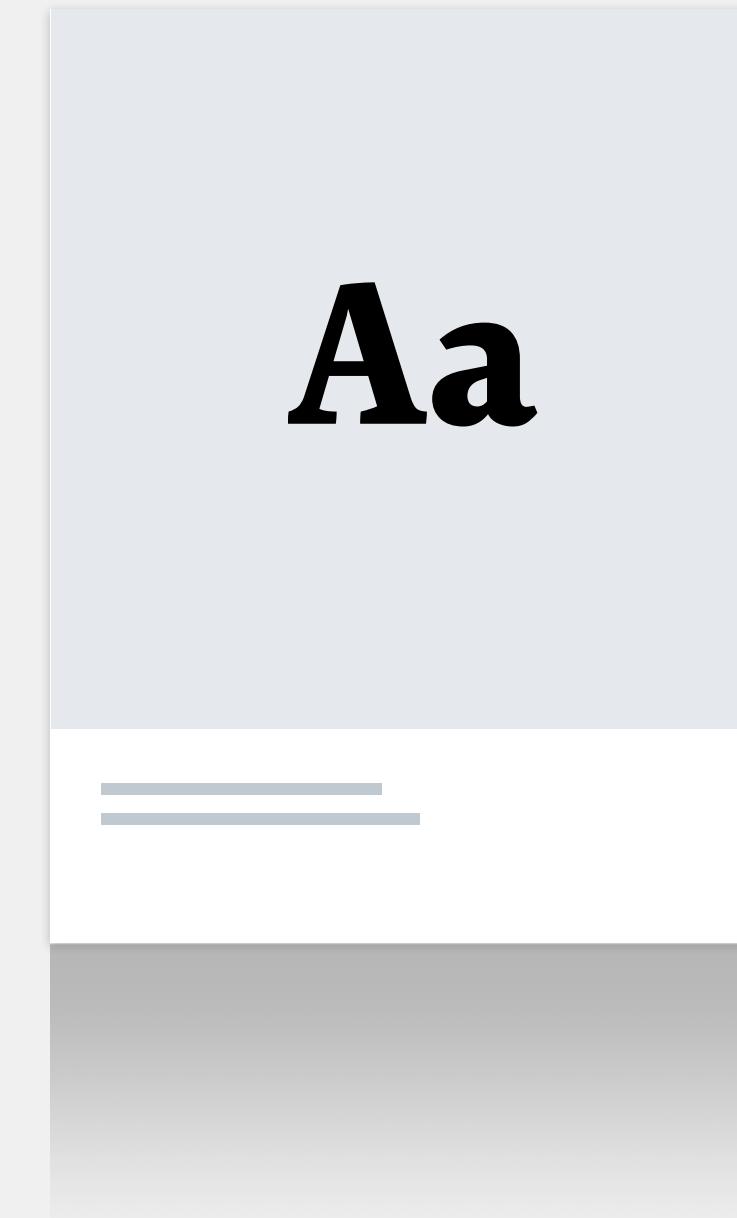
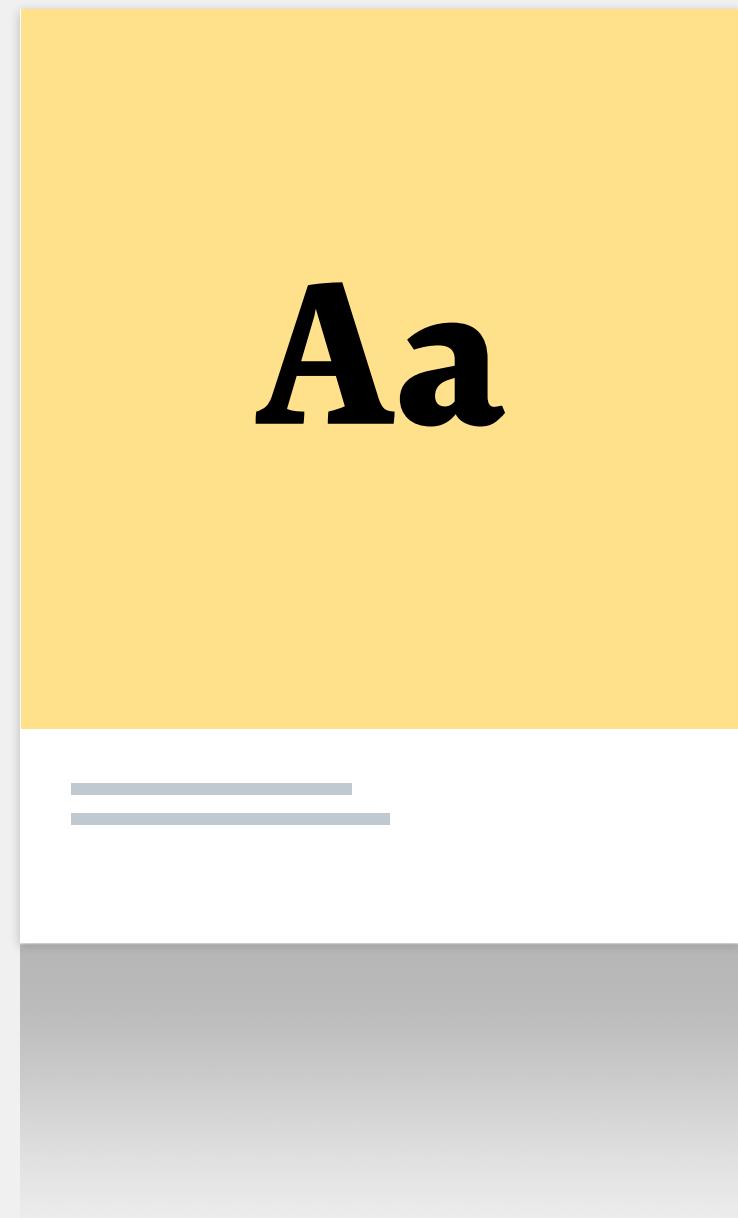
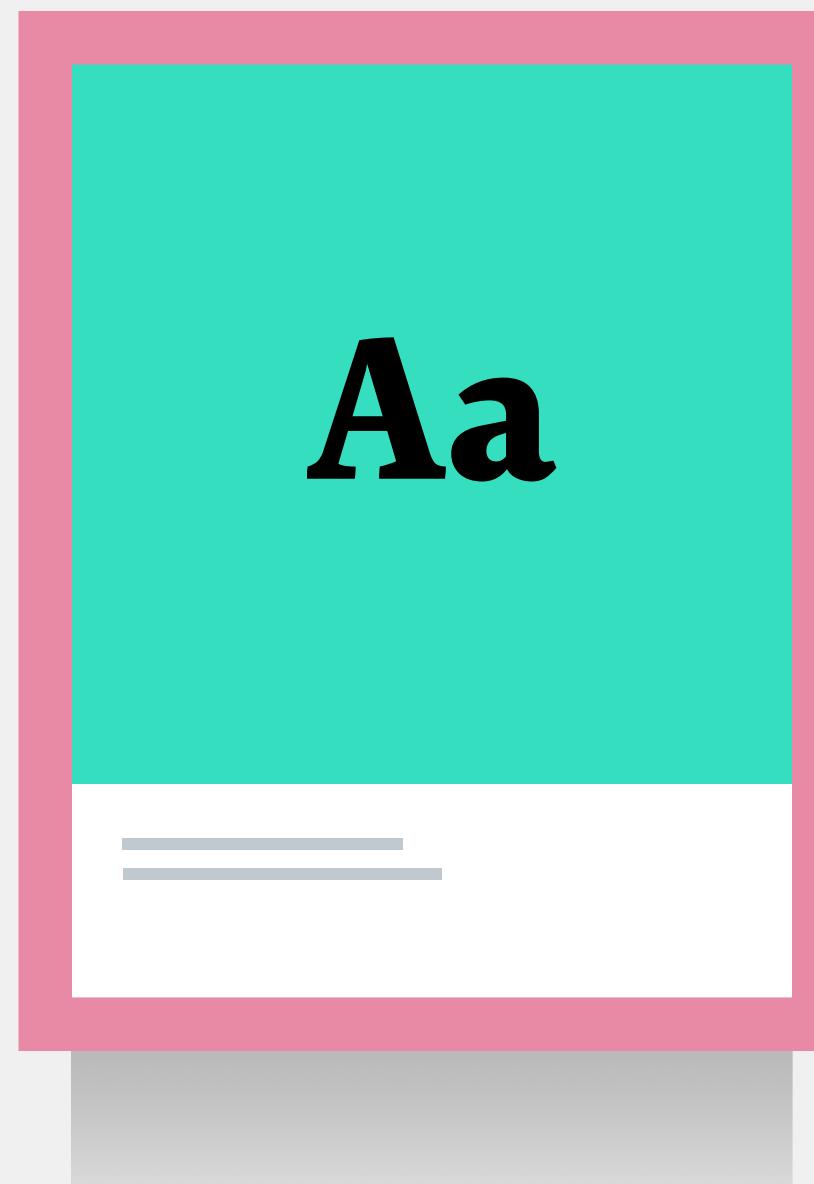
// Exports
export { localLogin, jwtLogin };
```



Friday night
deployment?!

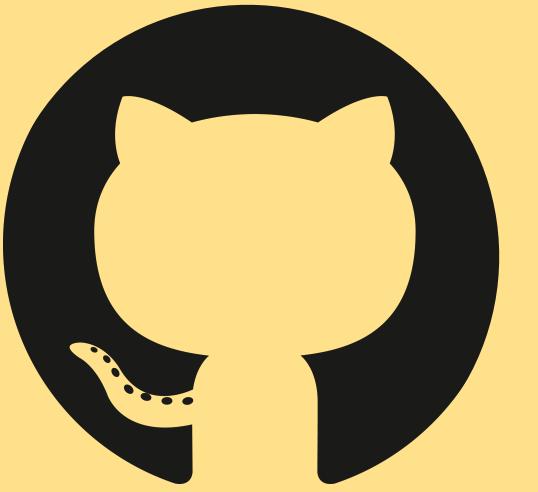
Color Contrast Automation

AA || AAA



$O(n^2)$

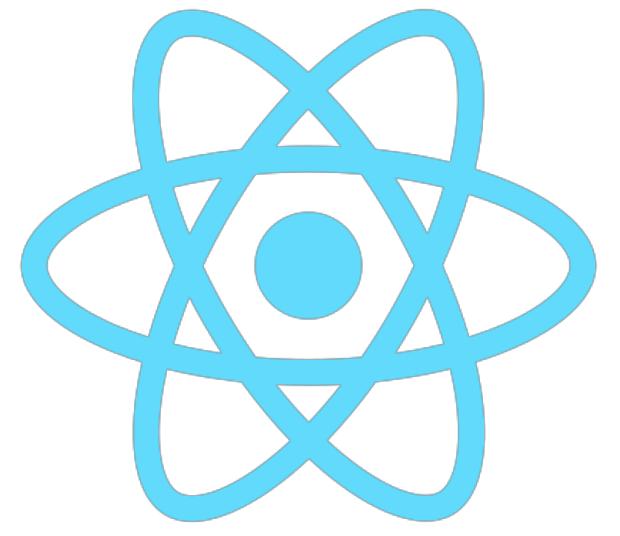
2500



Forward

A11y color contrast automation

github.com/palmaswell/forward



github.com/palmaswell/forward

Guideline 1.4 Distinguishable:

Make it easier for users to see and hear content including separating foreground from background.

AA >= 4.5:1

AAA >= 7:1

Large-scale text

AA >= 3:1

Large-scale text

AAA >= 4.5:1

Large-scale text

24px || 19px bold

Other exceptions

Logos and brand

Other exceptions

Inactive UI elements

Color contrast

Can you read this?

1.6 ratio

Color contrast

Can you read this?

8.05 ratio AAA

WCAG definition of relative luminance

Note 1: For the sRGB colorspace, the relative luminance of a color is defined as $L = 0.2126 * R + 0.7152 * G + 0.0722 * B$ where R, G and B are defined as:

if $RsRGB \leq 0.03928$ then $R = RsRGB/12.92$ else $R = ((RsRGB+0.055)/1.055) \wedge 2.4$

if $GsRGB \leq 0.03928$ then $G = GsRGB/12.92$ else $G = ((GsRGB+0.055)/1.055) \wedge 2.4$

if $BsRGB \leq 0.03928$ then $B = BsRGB/12.92$ else $B = ((BsRGB+0.055)/1.055) \wedge 2.4$

Excuse me?



Luminance

Relative Luminance

0

1

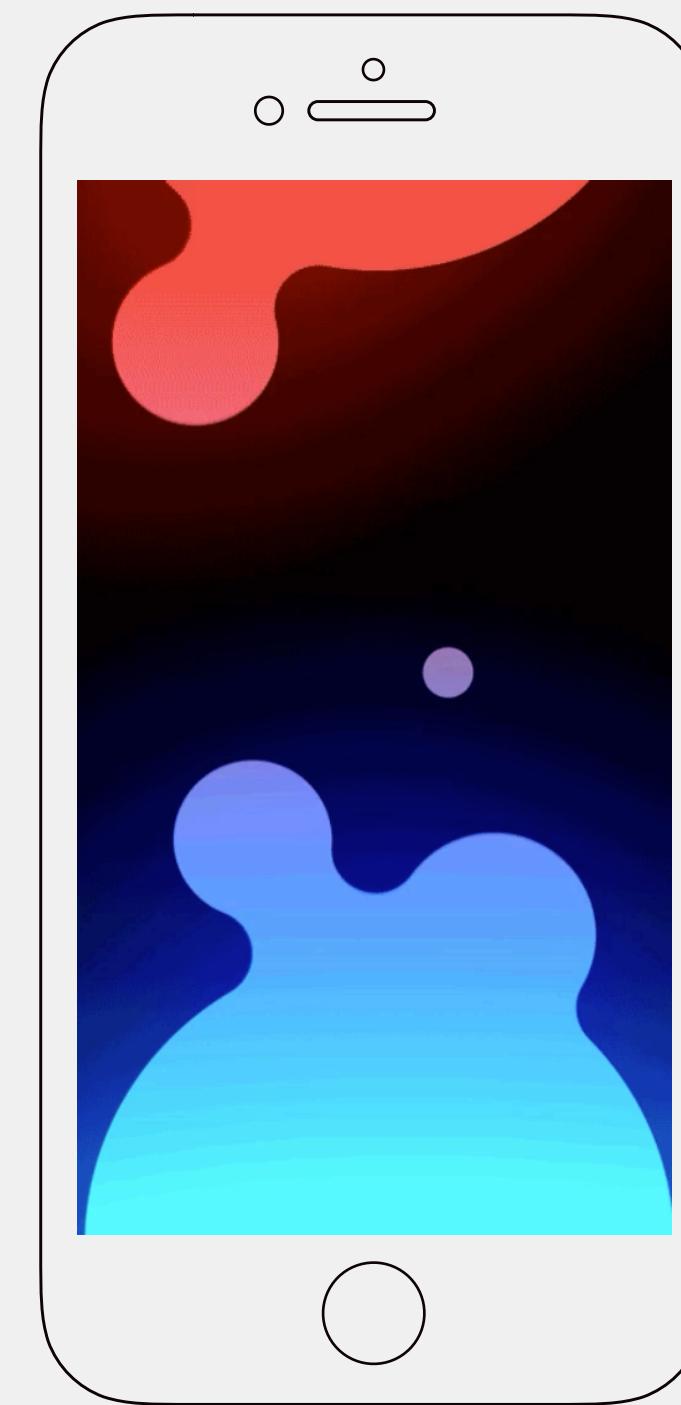
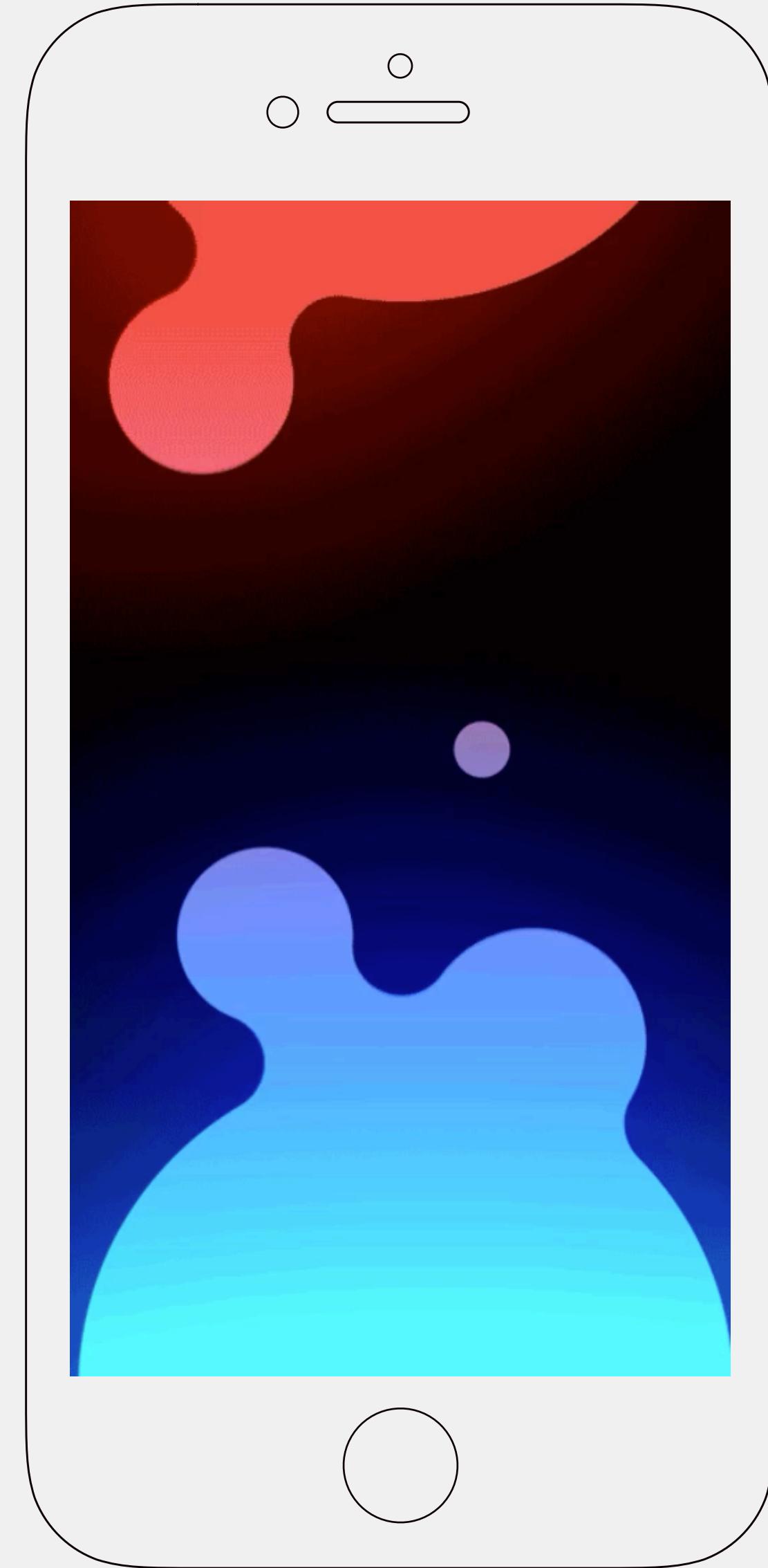
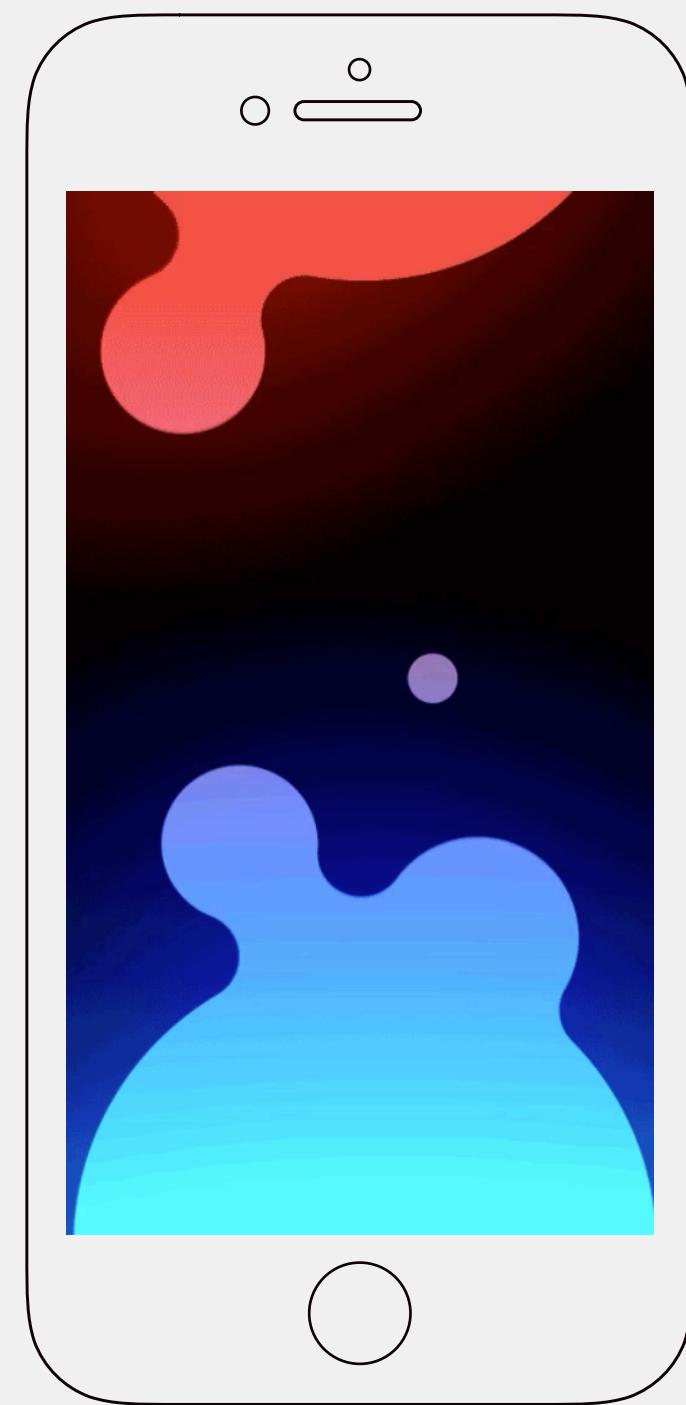
WCAG definition of relative luminance

Note 1: For the sRGB color space, the relative luminance of a color is defined as $L = 0.2126 * R + 0.7152 * G + 0.0722 * B$ where R, G and B are defined as:



sRGB
standardized by the
International Electrotechnical
Commission

sRGB



Our visual system



Web Browsers

**Default
color space**

Visual System

**Works in a
similar way**

Chromaticity

Chromaticity	Red	Green	Blue	White point
x	0.6400	0.3000	0.1500	0.3127
y	0.3300	0.6000	0.0600	0.3290
Y	0.2126	0.7152	0.0722	1.0000

Step 1

xyY color space

Y values represent the luminance of a color entry

```
export enum YValues {  
    r = 0.2126,  
    g = 0.7152,  
    b = 0.0722,  
}
```

Step 2

Color contrast

```
export function calculateRGBEntry(
  entry: number,
  y: Type.YValues): number {
  const average = entry / 255;

  return average <= 0.03928
    ? average / 12.92 * y
    : ((average + 0.055) / 1.055 ) ** 2.4 * y;
}
```

Step 2

Color contrast

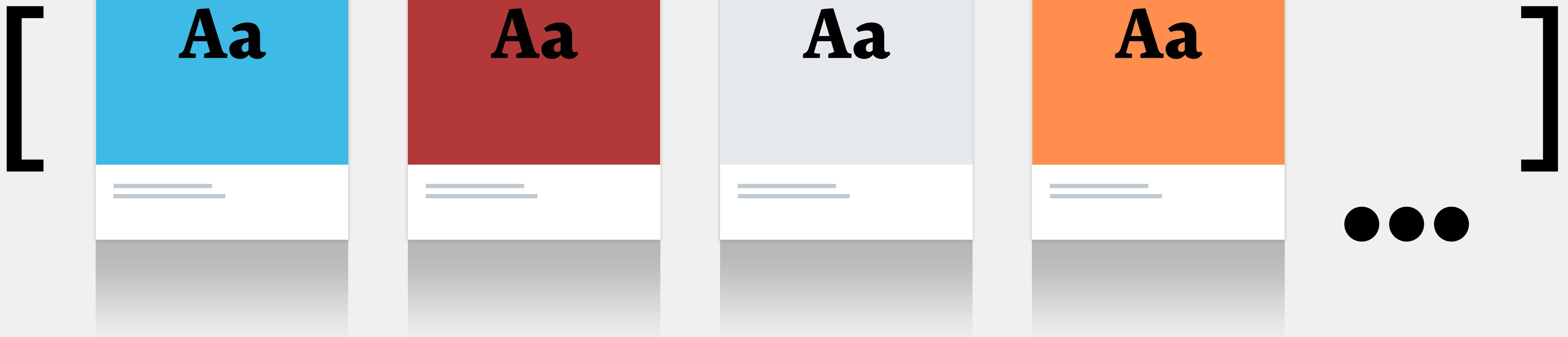
```
export function luminance(  
  sRGB: Type.RGB  
): number {  
  return calculateRGBEntry(sRGB[0], Type.YValues.r)  
    + calculateRGBEntry(sRGB[1], Type.YValues.g)  
    + calculateRGBEntry(sRGB[2], Type.YValues.b);  
}
```

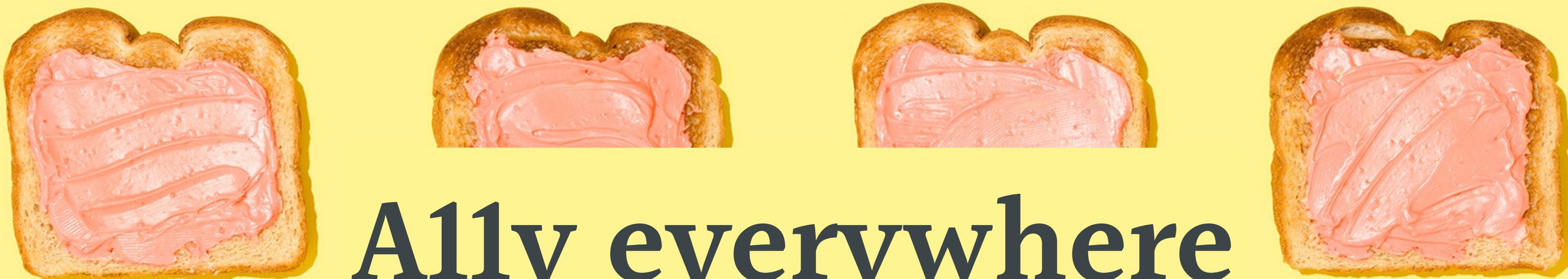
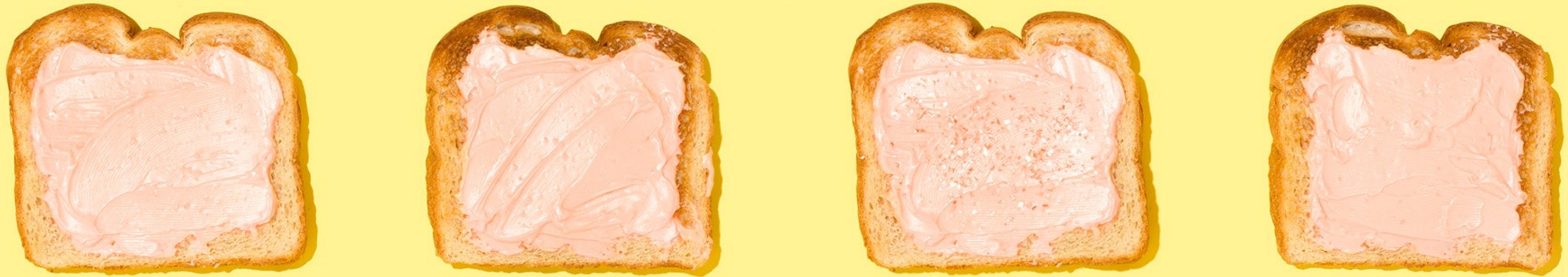
Step 2

Color contrast

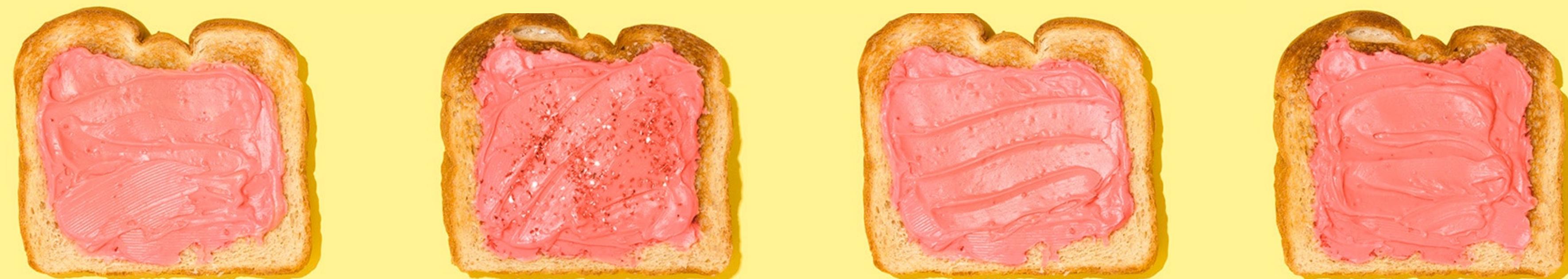
```
export function contrastRatio(  
  sRGB: Type.RGB,  
  sRGB2: Type.RGB): number {  
  const light = Math.max(luminance(sRGB), luminance(sRGB2));  
  const dark = Math.min(luminance(sRGB), luminance(sRGB2));  
  
  return +((light + 0.05) / (dark + 0.05)).toFixed(2);  
}
```

N colors





A11y everywhere



Quicksort

fast sorting algorithm



Step 3

Sort

```
export function quickSort(
  arr: Type.Color[],
  cb: (sRGB: Type.RGB) => number,
  lo: number,
  hi: number
): void {
  if (lo < hi) {
    const pivot = cb(arr[Math.floor((lo + hi) / 2)].rgb);
    const p = partition(arr, lo, hi, pivot, cb);
    quickSort(arr, cb, lo, p.hi);
    quickSort(arr, cb, p.lo, hi);
  }
}

export function sort(
  arr: Type.Color[],
  cb: (sRGB: Type.RGB) => number) {
  return quickSort(arr, cb, 0, arr.length - 1);
}
```

Step 3

Sort

```
export function quickSort(  
    arr: Type.Color[],  
    cb: (sRGB: Type.RGB) => number,  
    lo: number,  
    hi: number  
): void {  
    if (lo < hi) {  
        const pivot = cb(arr[Math.floor((lo + hi) / 2)].rgb);  
        const p = partition(arr, lo, hi, pivot, cb);  
        quickSort(arr, cb, lo, p.hi);  
        quickSort(arr, cb, p.lo, hi);  
    }  
}  
  
export function sort(  
    arr: Type.Color[],  
    cb: (sRGB: Type.RGB) => number) {  
    return quickSort(arr, cb, 0, arr.length - 1);  
}
```

Step 3

Sort

```
export function quickSort(
  arr: Type.Color[],
  cb: (sRGB: Type.RGB) => number,
  lo: number,
  hi: number
): void {
  if (lo < hi) {
    const pivot = cb(arr[Math.floor((lo + hi) / 2)].rgb);
    const p = partition(arr, lo, hi, pivot, cb);
    quickSort(arr, cb, lo, p.hi);
    quickSort(arr, cb, p.lo, hi);
  }
}

export function sort(
  arr: Type.Color[],
  cb: (sRGB: Type.RGB) => number) {
  return quickSort(arr, cb, 0, arr.length - 1);
}
```

Step 3

Sort

```
export function partition(
  arr: Type.Color[],
  lo: number,
  hi: number,
  pivot: number,
  cb: (sRGB: Type.RGB) => number): { lo: number, hi: number } {
  if (lo <= hi) {
    if (cb(arr[lo].rgb) < pivot) {
      return partition(arr, lo + 1, hi, pivot, cb);
    }
    if (cb(arr[hi].rgb) > pivot) {
      return partition(arr, lo, hi - 1, pivot, cb);
    }
    if (lo <= hi) {
      swap(arr, lo, hi);
      return partition(arr, lo + 1, hi - 1, pivot, cb);
    }
  }
  return { lo, hi };
}
```

Step 3

Sort

```
export function quickSort(  
    arr: Type.Color[],  
    cb: (sRGB: Type.RGB) => number,  
    lo: number,  
    hi: number  
): void {  
    if (lo < hi) {  
        const pivot = cb(arr[Math.floor((lo + hi) / 2)].rgb);  
        const p = partition(arr, lo, hi, pivot, cb);  
        quickSort(arr, cb, lo, p.hi);  
        quickSort(arr, cb, p.lo, hi);  
    }  
}  
  
export function sort(  
    arr: Type.Color[],  
    cb: (sRGB: Type.RGB) => number) {  
    return quickSort(arr, cb, 0, arr.length - 1);  
}
```

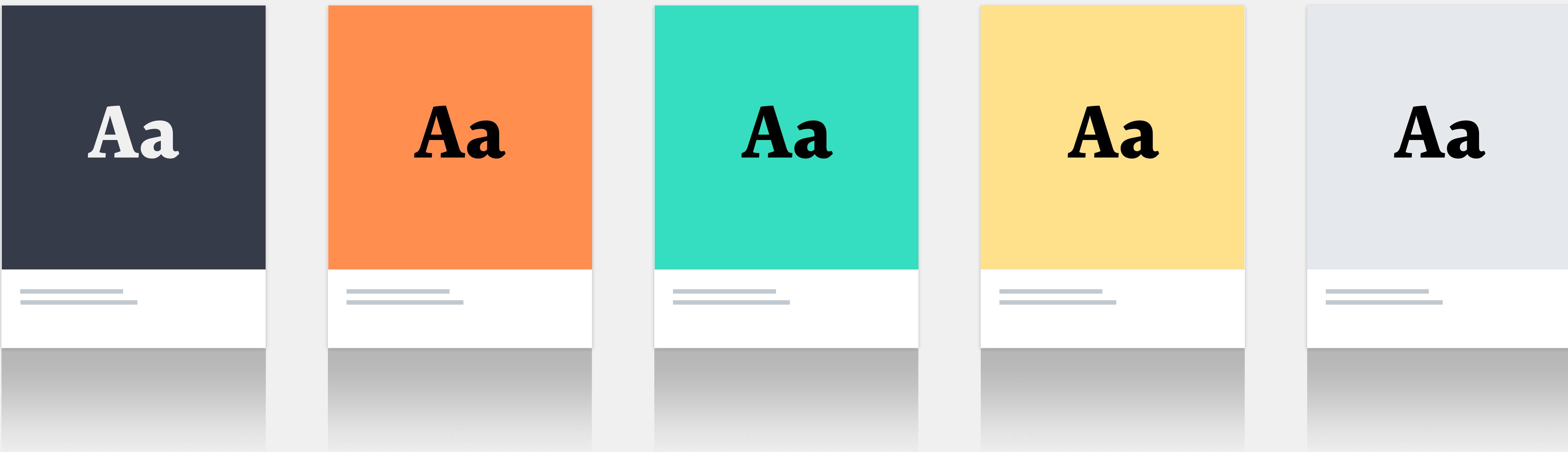
Step 3

Quicksort



Step 3

Sorted by luminance





Binary Search

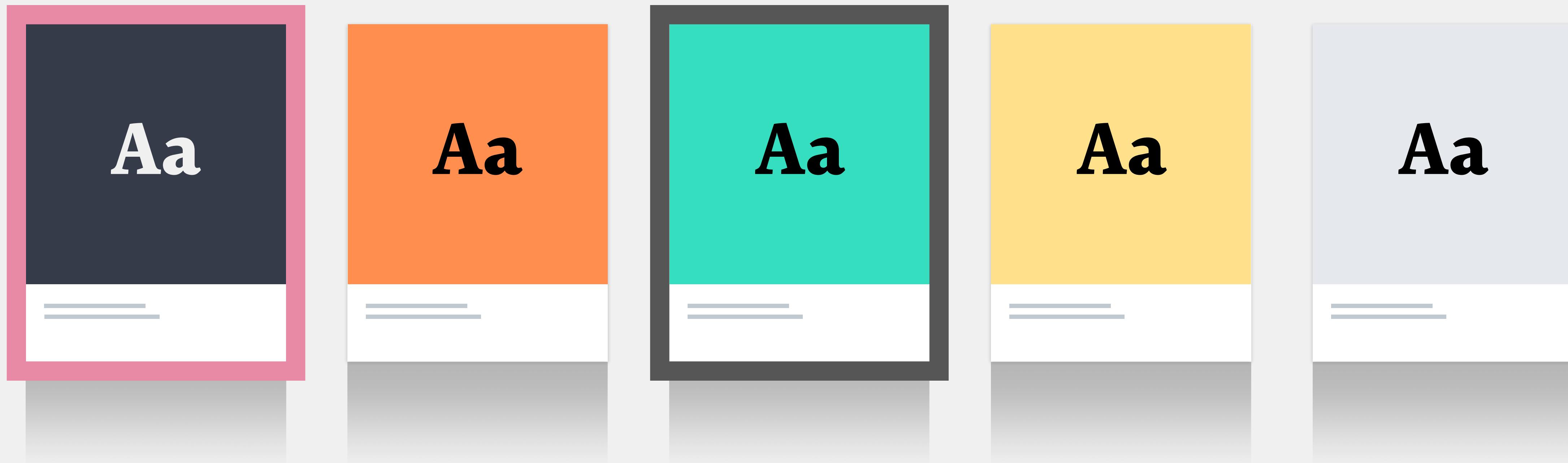
efficient algorithm that searches
a sorted list

Step 4

Binary Search

AAA color contrast check

Match!



Step 4

Search

```
export function search(
  arr: Type.Color[],
  el: Type.Color,
  val: Type.A11yRatio,
  type?: Type.Search): number | [] {
  if (type === Type.Search.upper) {
    return upperSearch(arr, el, val, 0, arr.length);
  }
  return lowerSearch(arr, el, val, 0, arr.length);
};
```

```
function lowerSearch<T extends QuickSearch>(
  arr: Array<T>,
  el: T,
  val: number,
  lo: number,
  hi: number): number | [] {
  const mid = Math.floor((lo + hi) / 2);
  const prev = mid - 1;
  const next = mid + 1;
  const midRatio = contrastRatio(arr[mid].rgb, el.rgb);
  if (lo < hi) {
    if (midRatio > val && prev >= 0) {
      if (contrastRatio(arr[prev].rgb, el.rgb) < val) {
        return mid;
      }
      return lowerSearch(arr, el, val, lo, mid);
    }
    if (midRatio < val && next < arr.length) {
      if (contrastRatio(arr[next].rgb, el.rgb) > val) {
        return next;
      }
      return lowerSearch(arr, el, val, mid, hi);
    }
    if (mid === val) {
      return mid;
    }
  }
  return [];
};
```

Step 4

Search

```
export function search(
  arr: Type.Color[],
  el: Type.Color,
  val: Type.AllRatio,
  type?: Type.Search): number | [] {
  if (type === Type.Search.upper) {
    return upperSearch(arr, el, val, 0, arr.length);
  }
  return lowerSearch(arr, el, val, 0, arr.length);
};
```

```
function lowerSearch<T extends QuickSearch>(
  arr: Array<T>,
  el: T,
  val: number,
  lo: number,
  hi: number): number | [] {
  const mid = Math.floor((lo + hi) / 2);
  const prev = mid - 1;
  const next = mid + 1;
  const midRatio = contrastRatio(arr[mid].rgb, el.rgb);
  if (lo < hi) {
    if (midRatio > val && prev >= 0) {
      if (contrastRatio(arr[prev].rgb, el.rgb) < val) {
        return mid;
      }
      return lowerSearch(arr, el, val, lo, mid);
    }
    if (midRatio < val && next < arr.length) {
      if (contrastRatio(arr[next].rgb, el.rgb) > val) {
        return next;
      }
      return lowerSearch(arr, el, val, mid, hi);
    }
    if (mid === val) {
      return mid;
    }
  }
  return [];
};
```

Step 4

Search

```
export function search(
  arr: Type.Color[],
  el: Type.Color,
  val: Type.A11yRatio,
  type?: Type.Search): number | [] {
  if (type === Type.Search.upper) {
    return upperSearch(arr, el, val, 0, arr.length);
  }
  return lowerSearch(arr, el, val, 0, arr.length);
};
```

```
function lowerSearch<T extends QuickSearch>(
  arr: Array<T>,
  el: T,
  val: number,
  lo: number,
  hi: number): number | [] {
  const mid = Math.floor((lo + hi) / 2);
  const prev = mid - 1;
  const next = mid + 1;
  const midRatio = contrastRatio(arr[mid].rgb, el.rgb);
  if (lo < hi) {
    if (midRatio > val && prev >= 0) {
      if (contrastRatio(arr[prev].rgb, el.rgb) < val) {
        return mid;
      }
      return lowerSearch(arr, el, val, lo, mid);
    }
    if (midRatio < val && next < arr.length) {
      if (contrastRatio(arr[next].rgb, el.rgb) > val) {
        return next;
      }
      return lowerSearch(arr, el, val, mid, hi);
    }
    if (mid === val) {
      return mid;
    }
  }
  return [];
};
```



Enhanced
Color

Step 5

Enhanced colors

```
export function createEnhanced(
    rawColor: Type.Color,
    aaa: Type.Color[] | [],
    aa: Type.Color[] | []
): Type.colorEnhanced {
    const rgb = rawColor.rgb;
    return {
        name: rawColor.name,
        rgb,
        aaa,
        aa,
        toRGB(): string {
            return toRGBString(rgb);
        },
        toHEX(): string {
            return tinyColor(toRGBString(rgb)).toHexString();
        },
        toHSL(): string {
            return tinyColor(toRGBString(rgb)).toHslString();
        },
        toRGBA(alpha: number): string {
            const color = tinyColor(toRGBString(rgb));
            color.setAlpha(alpha);
            return color.toRgbString()
        }
    }
}
```

Step 5

Enhanced colors

```
export function createEnhanced(
    rawColor: Type.Color,
    aaa: Type.Color[] | [],
    aa: Type.Color[] | []
): Type.colorEnhanced {
    const rgb = rawColor.rgb;
    return {
        name: rawColor.name,
        rgb,
        aaa,
        aa,
        toRGB(): string {
            return toRGBString(rgb);
        },
        toHEX(): string {
            return tinyColor(toRGBString(rgb)).toHexString();
        },
        toHSL(): string {
            return tinyColor(toRGBString(rgb)).toHslString();
        },
        toRGBA(alpha: number): string {
            const color = tinyColor(toRGBString(rgb));
            color.setAlpha(alpha);
            return color.toRgbString()
        }
    }
}
```



Colors
Under
Control

Demo



It's open

Use it, play with it, make it better!

github.com/palmaswell/forward



Benefits

- ⚡ New opportunities
- ⚡ Offer utility and elegance
- ⚡ Profitable
- ⚡ Legal compliance





Ally costs factors

- ⚡ Before: 10% (imaginary number)
- ⚡ Middle: x100%
- ⚡ After: x1000%

Questions?

swooped down to snap up the feed in a flurry of feathers and clucking, my grandmother would smile as I clapped and shrieked with delight.

She's a special woman, one who raised me almost as much as my parents did. Two Christmases ago, my grandmother became a bit more special to me, handing me a humble-looking present, a tiny bundle wrapped up in newsprint, possibly from that week's paper, with my name written on the adhesive tag stuck on the top. After a few seconds of wrestling with the manifold layers of newspaper—my grandmother can wrap, people—I managed to uncover a stack of three small books, each battered and worn.

I opened the topmost book, and tucked inside was an index card, covered with my grandmother's impeccable cursive writing. The note said these books were her father's diaries and that she wanted me to have them.

The three diaries are small, fragile-looking things, each bound in leather and well-used but not brittle—the newest from nearly ten years before my grandmother was born, the oldest from 1884, a full two decades before that. Some mold crept onto a few of the pages in the oldest diary, and its cover is beginning to flake, but that's the worst of the damage; they're in remarkable shape. I sat there, slowly turning them over in my hands, unable to speak.

After a moment . . .

my great-grandfather, a farmer, whose daughter grew up to follow in his footsteps.

But it's the heart of the diaries I cherish most, the blank pages my great-grandfather filled with his own writing. Each day has nine thin rules dedicated to it, three days per tiny page, each pair of pages spanning from Monday in the top left to Saturday on the bottom right. (Presumably, Sundays were reserved for a different kind of book.) Each day contained a brief phrase about the weather ("Cloudy and v. cold," "Pleasant + warm") with one or two significant events ("Marvin and I drained ice from Center Pond," "Got the sleigh shod"). A few spare words in a sloping, schoolboy's hand doggedly filled in each day over the course of decades.

My grandmother and I talk often, or as often as I remember to call. She makes a show of complaining when she hasn't heard from this grandchild or that in a few days, always with a smile in her voice as she acts mock-aggrieved. She doesn't cook as often as she used to. In fact, a task can leave her tired for days, whether it's a half hour of peeling potatoes, a short drive to the market, or simply a few visitors stopping by for an afternoon chat. A small item on her schedule is an event.

Thanks!

Github: [@palmaswell](#)

Twitter: [@palmaswell](#)

mauricio.palma@sinnerschrader.com