

Custom CSS is the path to  
inconsistent UI

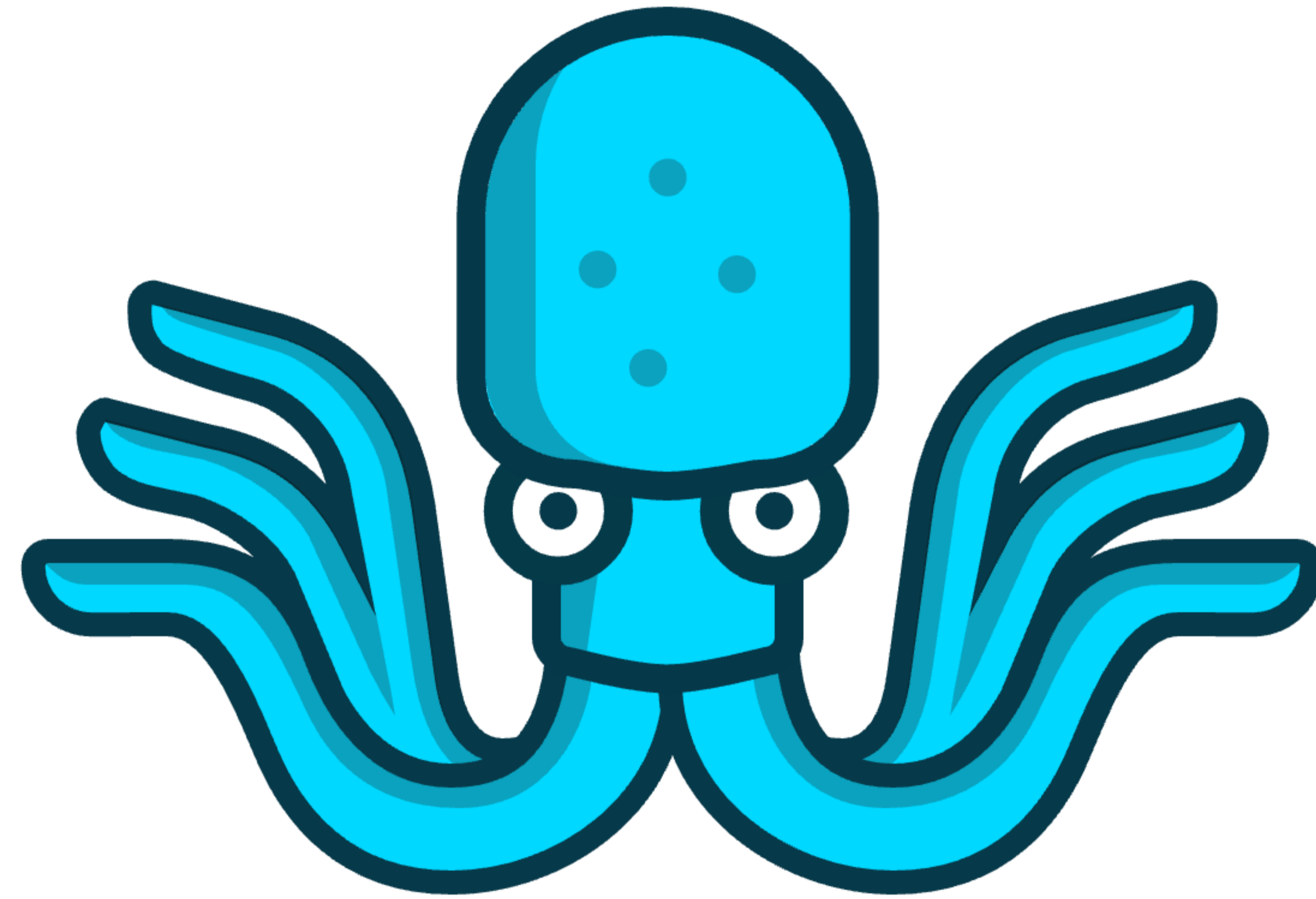
**Artem Sapegin**

[@iamsapegin](#)

Wayfair

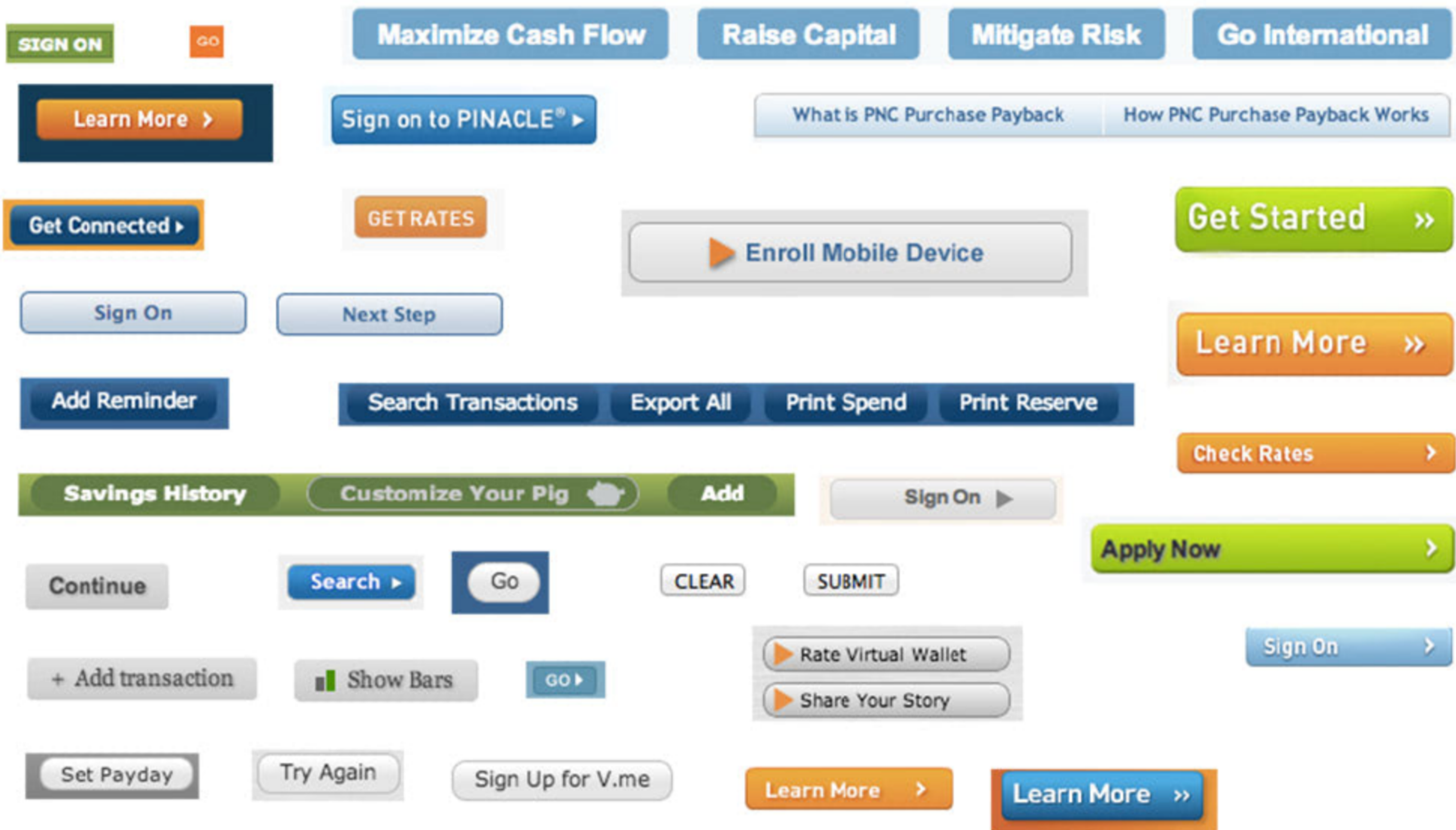
Berlin, Germany





[react-styleguidist.js.org](https://react-styleguidist.js.org)

What's the issue?



## Headers

h5, .header5

Open Sans Light 32px

h1, .header1

Open Sans Regular 24px

h8, .header8

Open Sans Light 24px

h20, .header20

Open Sans Regular 22px

h2, .header2

Open Sans Regular 20px

h16, .header16

Open Sans Regular 20px

h3, .header3a

Open Sans Regular 18px

h19, .header19

Open Sans Regular 18px

h21, .header21

Open Sans Regular 18px

h4, .header4

Open Sans Regular 16px

h4, .header4semibold

Open Sans Semibold 16px

h3, .header3b

Open Sans Regular 15px

h14, .header14

Open Sans Regular 15px

h7, .header7

Open Sans Semibold 14px

h11, .header11

Open Sans Regular 14px

h17, .header17

Open Sans Regular 14px

h18, .header18

Open Sans Regular 14px

h9, .header9

Open Sans Semibold 12px

h10, .header10

Open Sans Semibold 12px

## Paragraph

p

Open Sans Regular 13px

.p-semibold

Open Sans Semibold 13px

.p-aaa

Open Sans Regular 13px

.p-aaa-semibold

Open Sans Semibold 13px

.p-fff

Open Sans Regular 13px

.p-fff-semibold

Open Sans Semibold 13px

.p-link-semibold

Open Sans Semibold 13px

.p-highlight

Open Sans Regular 13px

.p-strike

Open Sans Semibold 13px

.p-red

Open Sans Regular 13px

.p-red-semibold  
red asterisk

Open Sans Semibold 13px

## Small

small

Open Sans Regular 11px

small-666

Open Sans Regular 11px

small-red

Open Sans Semibold 11px

small-green

Open Sans Semibold 11px

small-white-bold

Open Sans Bold 11px

small-white

Open Sans Regular 11px

## Other

hint-text-20

Open Sans Regular 20px

hint-text-15

Open Sans Regular 15px

new-item

Open Sans Regular 14px

hint-text-13

Open Sans Regular 13px

orange asterisk

Open Sans Semibold 13px

field-label

OPEN SANS REGULAR 11PX

progress-header

OPEN SANS REGULAR 11PX

tile-status

OPEN SANS BOLD 11PX

footnote

Open Sans Semibold 10px

lab-order-header

Open Sans Semibold 8px

lab-order-label

OPEN SANS SEMIBOLD 8PX

monospaced-font

Droid sans mono 13PX

## Lead

.lead

Open Sans Regular 15px

.lead-333

Open Sans Regular 15px

.lead-red

Open Sans Regular 15px

.lead-green

Open Sans Regular 15px



# Sources of custom CSS

- Typography
- Whitespace & layout



# Custom CSS

```
.description {  
  margin-bottom: 20px;  
  font-size: 16px;  
}
```

```
.count {  
  margin-left: 0.75em;  
  font-size: 14px;  
  opacity: 0.5;  
}
```

CSS is too  
expressive

Can we fix that?

Design tokens

Styleguide — Marvel

https://marvelapp.com/styleguide/design/color-scheme

We're hiring

**Marvel**

STYLEGUIDE

Overview

Design

Color Scheme

Typography

Animation

Border Radius

Box Shadow

Layout

Icons COMING SOON

Components

Utility

GitHub

Discord

Twitter

# Color Scheme

## Brand colors

Primary brand colors are used for elements that must reflect Marvel's brand. Each color has a darker and a lighter shade.

Light Marvel #85D7FF	Light Purple #A389F4	Light Pink #FF7CE5	Light Orange #FF9E7C
Marvel #1FB6FF	Purple #7E5BEF	Pink #FF49DB	Orange #FF7849
Dark Marvel #009EEB	Dark Purple #592DEA	Dark Pink #FF16D1	Dark Orange #FF5216
Light Green #29EB7F	Light Yellow #FFD55F		
Green #13CE66	Yellow #FFC82C		
Dark Green #0F9F4F	Dark Yellow #F8B700		

cf-ui styleguide

https://cloudflare.github.io/cf-ui/#cf-design-gradients

# Gradients

- cf-design
  - Colors
  - Gradients**
  - Icons
- cf-ui
  - Box
  - Button
  - Callout
  - Card
  - Card Builder
  - Checkbox
  - Code
  - Copyable Textarea
  - Dropdown
  - Dynamic Content
  - Form
  - Heading
  - Icon
  - Input
  - Label
  - Link
  - List
  - Loading
  - Modal
  - Notifications
  - Page
  - Pagination
  - Pagination Builder
  - Progress
  - Radio
  - Select
  - Table
  - Table Builder
  - Tabs

skyDew : `linear-gradient(left, #76C4E2, #85CBA8)`

twilightDew : `linear-gradient(left, #8176B5, #85CBA8)`

twilightSky : `linear-gradient(left, #8176B5, #76C4E2)`

twilightSunset : `linear-gradient(left, #8176B5, #BA77B1)`

twilightDawn : `linear-gradient(left, #8176B5, #F16975)`

## Font Sizes

**\$font-size-xxx-large** (64px)

**\$font-size-xx-large** (48px)

**\$font-size-x-large** (36px)

**\$font-size-large** (24px)

**\$font-size-base** (16px)

**\$font-size-small** (13px)

**\$font-size-x-small** (11px)

## Spacing

\$space-xx-large (128px)

\$space-x-large (64px)


\$space-large (48px)

\$space-base (24px)

\$space-small (16px)

\$space-x-small (8px)

\$space-xx-small (4px)



Search

- What's New
- Getting Started
- Platforms
- Design Guidelines
- Accessibility
- Component Blueprints
- Utilities
- Design Tokens**
- Icons
- Downloads
- Articles
- FAQ

Find Token Format: **Sass**

## Opacity

Use these opacity tokens for element transparency.

Token	Example	Released	Themeable	Support
<b>\$opacity-5</b> 50% transparency of an element	 0.5	1.0.0	No	<span>I</span>
<b>\$opacity-8</b> 80% transparency of an element	 0.8	1.0.0	No	<span>I</span>

## Line Height

Use these tokens for changing the line-height of elements. Usually, the line-height-text is already inherited by default. Only set it if you need to apply it again.

Token	Example	Released	Themeable	Support
<b>\$line-height-heading</b> Unitless line-heights for reusability	The quick brown fox jumps over the lazy dog 1.25	1.0.0	No	<span>GA</span>
<b>\$line-height-reset</b> Remove extra leading. Unitless line-heights for reusability	The quick brown fox jumps over the lazy dog 1	1.0.0	No	<span>GA</span>
<b>\$line-height-tab</b> Line heights for tabs	The quick brown fox jumps  over the lazy dog --	1.0.0	No	<span>GA</span>

### CATEGORIES

- Colors
- Background Color
- Text Color
- Border Color
- Font
- Font Size
- Opacity
- Line Height
- Spacing
- Radius
- Sizing
- Shadow
- Media Query
- Z-index



Design tokens are  
everything you use more  
than once in your styles

# Design tokens implementation

- Sass variables
- CSS custom properties
- JavaScript constants
- JSON

# Custom CSS & design tokens

```
.description {  
  margin-bottom: var(--spacing--3);  
  font-size: var(--fontSize--base);  
}
```

```
.count {  
  margin-left: var(--spacing--2);  
  font-size: var(--fontSize--small);  
  color: var(--color--light);  
}
```

Not all token  
combinations are good

# Bad token combinations

```
.cant__read_this {  
  font-size: var(--fontSize--x-small);  
  color: var(--color--light);  
}
```

Can you read me?!

Can we do  
better?

Stop writing CSS!

# Primitive components



- 1. Typography**
2. Spacing & layout

Text style - Shopify Polaris

https://polaris.shopify.com/components/titles-and-text/text-style

shopify

GUIDES CONTENT DESIGN COMPONENTS PATTERNS Search

- GET STARTED
- COMPONENT STATUS
- ACTIONS
- STRUCTURE
- IMAGES AND ICONS
- FEEDBACK INDICATORS
- TITLES AND TEXT
  - Caption
  - Display text
  - Heading
  - Subheading
  - Text container
  - Text style**
  - Examples
  - Best practices

# Text style

Text style enhances text with additional visual meaning. For example, using subdued text to de-emphasize it from its surrounding text.

Web Android iOS

## Examples

Subdued text style

Use to de-emphasize a piece of text that is less important to the merchant than other nearby text. May also be used to indicate when normal content is absent, for example, "No supplier listed". Don't use only for aesthetic effect.

No supplier listed

COPY CODE EDIT IN SANDBOX REACT HTML



## What's New

Component Status

Roadmap

## Design

Color

Tokens

Typography

## Guides

Getting Started

Theming

Styling

Import Syntax

Render Props

## Components

Avatar

Button

ButtonGroup

Card

Checkbox

Dialog

Dropdown

FormField

Icon

Layout

Link


# Examples

## Import Syntax #

```
import Text from 'mineral-ui/Text';
```

## Basic Usage

Text renders content with the correct styles and semantics. Text can include complex content, such as an [Icon](#), and can be rendered as any HTML element.

Lorem ipsum  dolor sit amet, consectetur adipiscing elit. Sed non purus id dolor maximus eleifend nec in nunc. Integer quis lacinia leo, in suscipit orci. Donec enim augue, vehicula vel enim a, vestibulum tempus nunc.

```
<Text>  
  Lorem ipsum <IconCloud size="large" /> dolor sit amet, consectetur  
  adipiscing elit. Sed non purus id dolor maximus eleifend nec in nunc.  
  Integer quis lacinia leo, in suscipit orci. Donec enim augue, vehicula vel  
  enim a, vestibulum tempus nunc.  
</Text>
```

# Text styles

- Normal text
- Secondary text
- **Error**
- etc.

Minimal number of styles  
for most of your use cases

# What else?

- Custom HTML element (p, span, label)
- Alignment (left, center)
- Whitespace (margin)

# API: selecting a style

- `<Text primary>` 😊
- `<Text secondary>` 😊
- `<Text primary secondary>` 😭

# API: selecting a style

- `<Text variant="primary">` 😊
- `<Text variant="secondary">` 😊



Make impossible states  
impossible

# API: rigidity vs flexibility

- `<Text variant="secondary">` 😊
- `<Text color="red" size="xs">` 😊

# API: custom HTML element

- `<Text>` → `<p>` 😊
- `<Text inline>` → `<span>` 😐
- `<Text label>` → `<label>` 😭
- `<Text as="label">` → `<label>` 😊

# Example

<Text>

The Saluki is classed as a sighthound and is typically deep-chested and long-legged. Salukis are “sight” hounds – hunting by sight – and run their quarry down to kill or retrieve it.

</Text>

<Text as="small" variant="secondary">

Don't leave any food on a table when saluki is around.

</Text>

The Saluki is classed as a sighthound and is typically deep-chested and long-legged. Salukis are “sight” hounds – hunting by sight – and run their quarry down to kill or retrieve it.

Don't leave any food on a table when saluki is around.

# Example

```
<Heading size={1}>Saluki</Heading>
```

```
<Text>
```

```
The Saluki is classed as a sighthound  
and is typically deep-chested and  
long-legged. Salukis are “sight”  
hounds – hunting by sight – and  
run their quarry down to kill  
or retrieve it.
```

```
</Text>
```

```
<Heading size={2}>Caveats</Heading>
```

```
<Text as="small" variant="secondary">
```

```
Don't leave any food on a table  
when saluki is around.
```

```
</Text>
```

## Saluki

The Saluki is classed as a sighthound and is typically deep-chested and long-legged. Salukis are “sight” hounds – hunting by sight – and run their quarry down to kill or retrieve it.

### Caveats

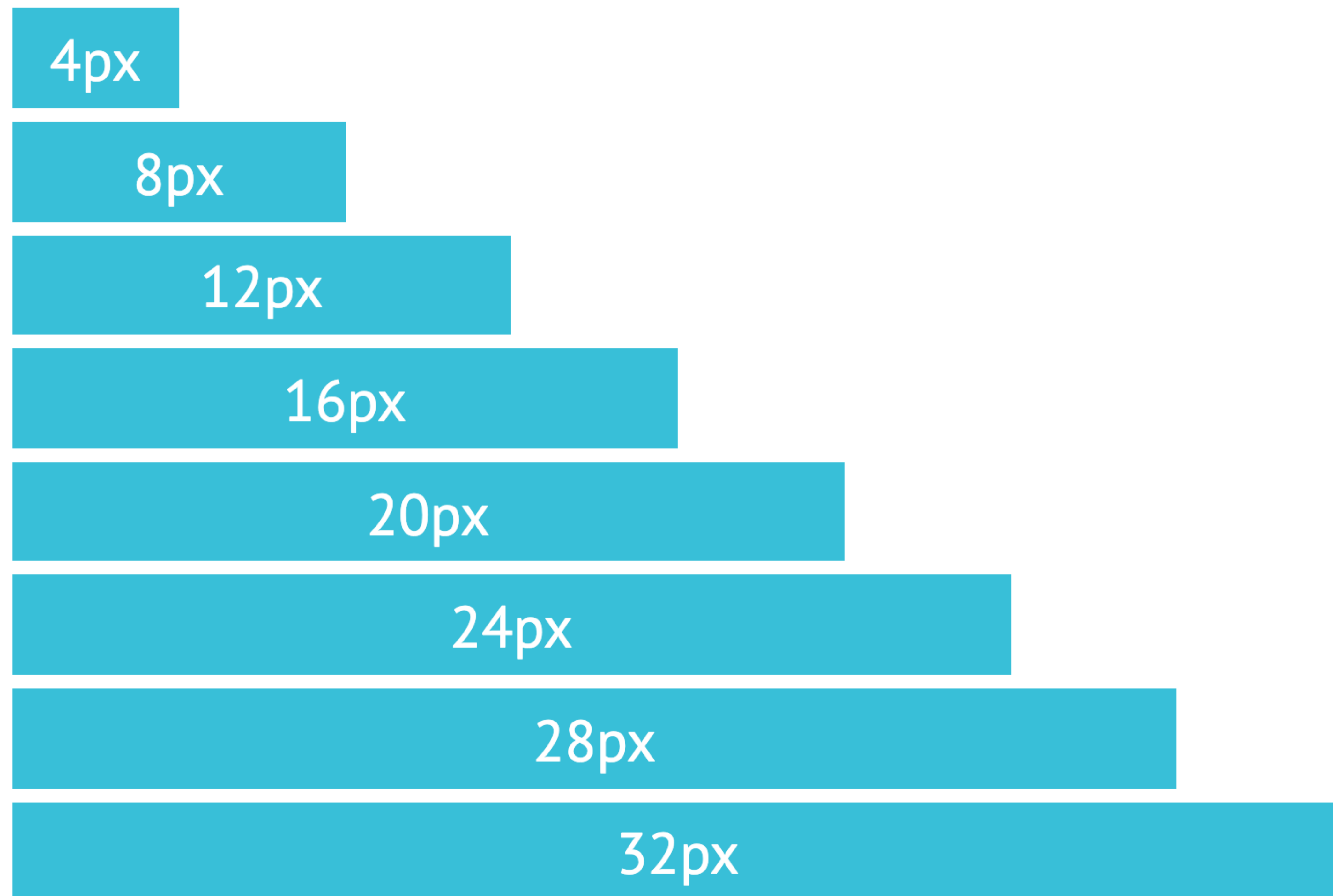
Don't leave any food on a table when saluki is around.

1. Typography
- 2. Spacing & layout**

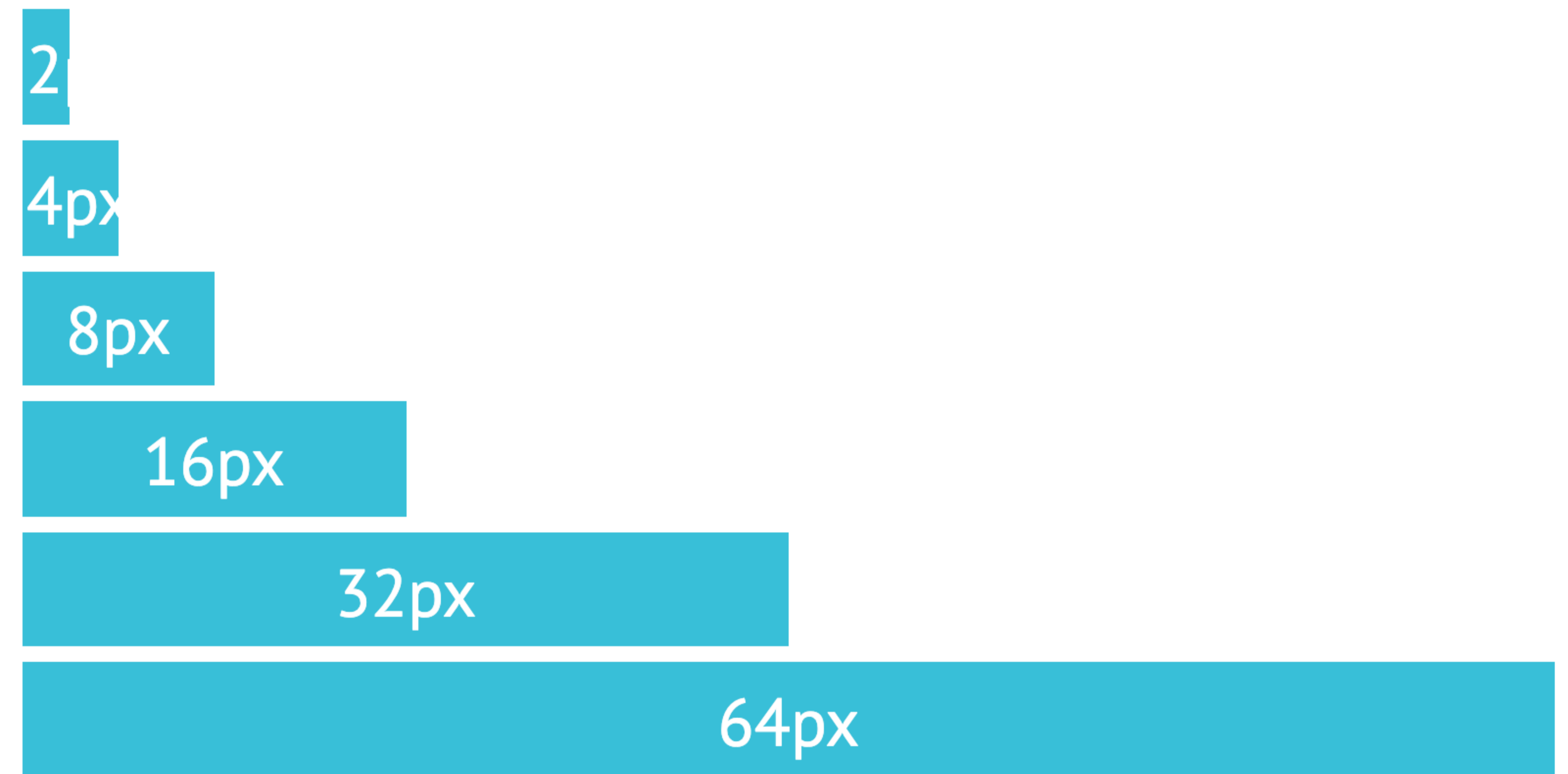


# Spacing scales

## Linear progression



## Geometric progression





# Scale naming

- x-small small medium large x-large
- xs sm md lg xl
- xs s m l xl
- 1 2 3 4 5
- ~~alpha beta gamma delta epsilon~~

# Spacing scale

```
const space = {  
  xxs: '2px',  
  xs: '4px',  
  s: '8px',  
  m: '16px',  
  l: '32px',  
  xl: '64px',  
  xxl: '128px'  
};
```

```
const space = {  
  xxs: '0.125rem',  
  xs: '0.25rem',  
  s: '0.5rem',  
  m: '1rem',  
  l: '2rem',  
  xl: '3rem',  
  xxl: '4rem'  
};
```

# Margins and padding

```
<Box mb="xl" as="section">  
  <Box mb="l">  
    <Heading level={1}>Saluki</Heading>  
  </Box>  
  <Text>The Saluki is classed...</Text>  
</Box>
```

## Saluki

The Saluki is classed as a sighthound and is typically deep-chested and long-legged. Salukis are “sight” hounds — hunting by sight — and run their quarry down to kill or retrieve it.

# Flexbox layouts

```
<Flex  
  alignItems="center"  
  justifyContent="center"  
  flexDirection="row"  
  flexWrap="wrap"  
>
```

# Flexbox layouts

```
<Flex as="form">
  <Box flexGrow={1}>
    <Input
      type="email"
      defaultValue=""
      placeholder="Email"
      aria-label="Email"
    />
  </Box>
  <Box pl="m">
    <Button variant="primary" type="submit">
      Subscribe
    </Button>
  </Box>
</Flex>
```

# Responsive layouts

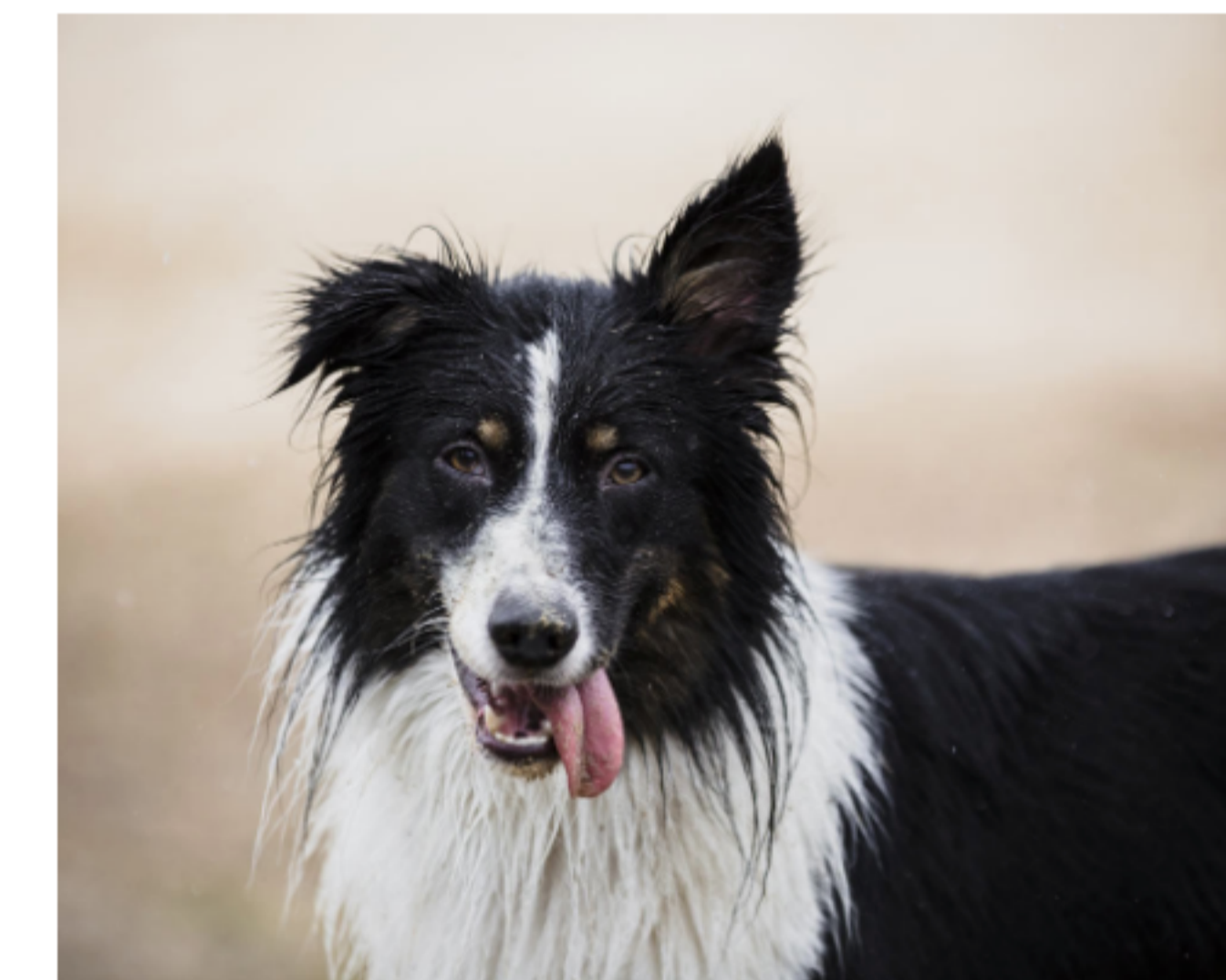
```
//          mobile  tablet  desktop
<Box
  width={[ 1,      1/2,    1/4    ]}
  m={[   's',    'm',    'l'    ]}
/>
```

# CSS Grid layouts

```
<Grid  
  gridGap="m"  
  gridColumn="..."  
  gridRow="..."  
  gridAutoFlow="..."  
  gridAutoRows="..."  
>
```

# CSS Grid layouts

```
<Grid gridGap="m"  
  gridTemplateColumns="  
    repeat(  
      auto-fit,  
      minmax(200px, 1fr)  
    )  
">  
  <Image src="dog1.jpg" alt="Dessi" />  
  <Image src="dog2.jpg" alt="Tsiri" />  
  <Image src="dog3.jpg" alt="Twiggy" />  
</Grid>
```





Implementation

# Plain CSS

```
import React from 'react';
import cx from 'clsx';

export default function Box({ m, mt, mr, mb, ml, p, pt, pr, pb, pl, ...props }) {
  return (
    <div
      className={cx('Box', {
        [`Box-margin--${m}`]: m,
        [`Box-marginTop--${mt}`]: mt,
        [`Box-marginRight--${mr}`]: mr,
        [`Box-marginBottom--${mb}`]: mb,
        [`Box-marginLeft--${ml}`]: ml,
        [`Box-padding--${p}`]: p,
        [`Box-paddingTop--${pt}`]: pt,
        [`Box-paddingRight--${pr}`]: pr,
        [`Box-paddingBottom--${pb}`]: pb,
        [`Box-paddingLeft--${pl}`]: pl
      })}
      {...props}
    />
  )
}
```

# Naïve CSS in JS

```
import styled from 'styled-components';
import theme from './theme';

const Box = styled.div`
  min-width: 0;
  margin: ${props => theme.space[props.m]};
  margin-left: ${props => theme.space[props.ml]};
  margin-top: ${props => theme.space[props.mt]};
  margin-right: ${props => theme.space[props.mr]};
  margin-bottom: ${props => theme.space[props.mb]};
  padding: ${props => theme.space[props.p]};
  padding-left: ${props => theme.space[props.pl]};
  padding-top: ${props => theme.space[props.pt]};
  padding-right: ${props => theme.space[props.pr]};
  padding-bottom: ${props => theme.space[props.pb]};
`;

export default Box;
```

# styled-system by Brent Jackson

- Build primitive UI components
- Control styles with props
- Design tokens
- Responsive props
- Margins, padding, colors, fonts, Flexbox, CSS Grid, etc.
- styled-components, Emotion, etc.

# styled-system

```
import styled from 'styled-components';  
import { space } from 'styled-system';  
  
export default styled.div`  
  min-width: 0;  
  ${space};  
`;  
;
```

Recap

# Why use primitives

- Consistency
- Discoverable API
- Faster development
- Code reuse
- Code reviews
- Documentation
- Accessibility
- Common language

CSS → Preprocessors →  
CSS in JS → Primitives





# Login X

Username:

Password:

[Forget password?](#)



[component-driven.io](https://component-driven.io)

# Libraries and examples

- Text
- Heading
- Box & Flex: Rebass Grid
- Grid (Stack): Stack Styled
- styled-system

# Thank you and use components!

Slides: [bit.ly/usecomponents](https://bit.ly/usecomponents)

Me: [sapegin.me](https://sapegin.me)

Twitter: [@iamsapegin](https://twitter.com/iamsapegin)



# Image credits

- Max Stoiber
- Priyanka Godbole
- Nathan Curtis
- Brad Frost