Writing Declarative React

A Styled Components Story

@zealigan

Eric Adamski

Software Developer at MB3
Twitter: @zealigan
Github: @ericadamski

Styled Components 💅

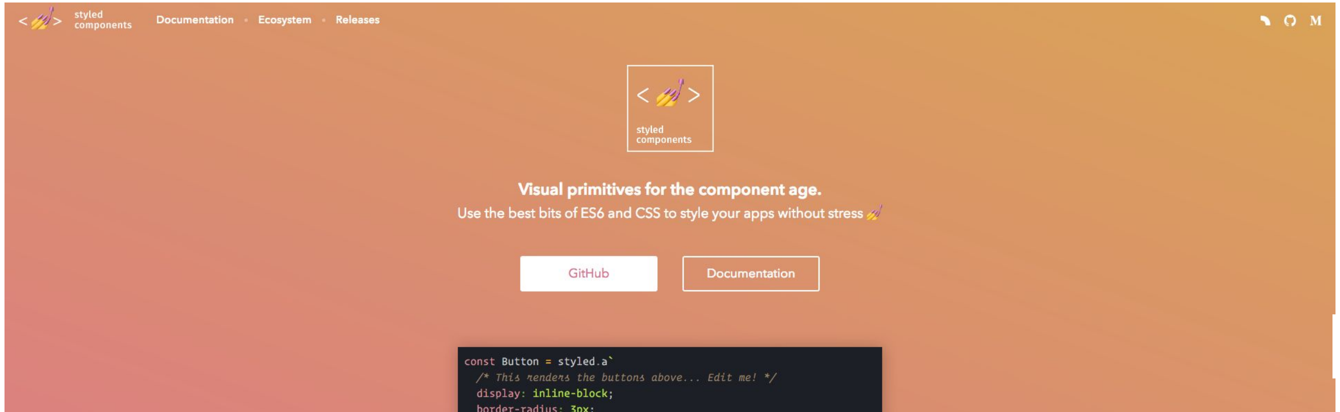
Has anyone ...

- Seen a talk on Styled Components?
- Used Styled Components?

Styled Components 💅

A CSS-in-JS library

In the most literal meaning



Visual primitives for the component age.

Use the best bits of ES6 and CSS to style your apps without stress 💅

```
const Button = styled.a`
  /* This renders the buttons above... Edit me! */
  display: inline-block;
  border-radius: 3px;
```

@zealigan

What are we talking about?

How Styled Components will help you reason about your React Components

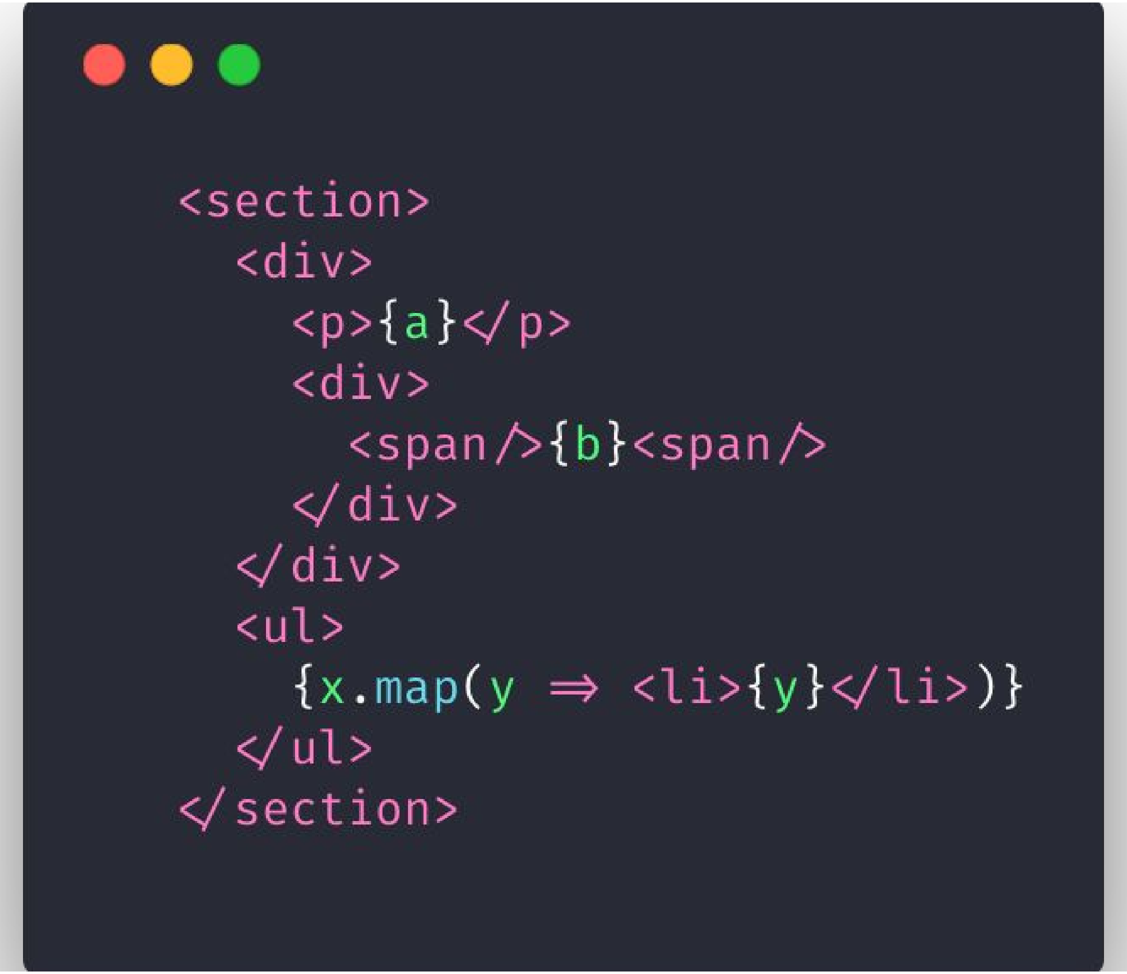
What are we NOT talking about?

- Why choose CSS-in-JS?
- Any other CSS-in-JS library
- How to write Styled Components

@zealigan

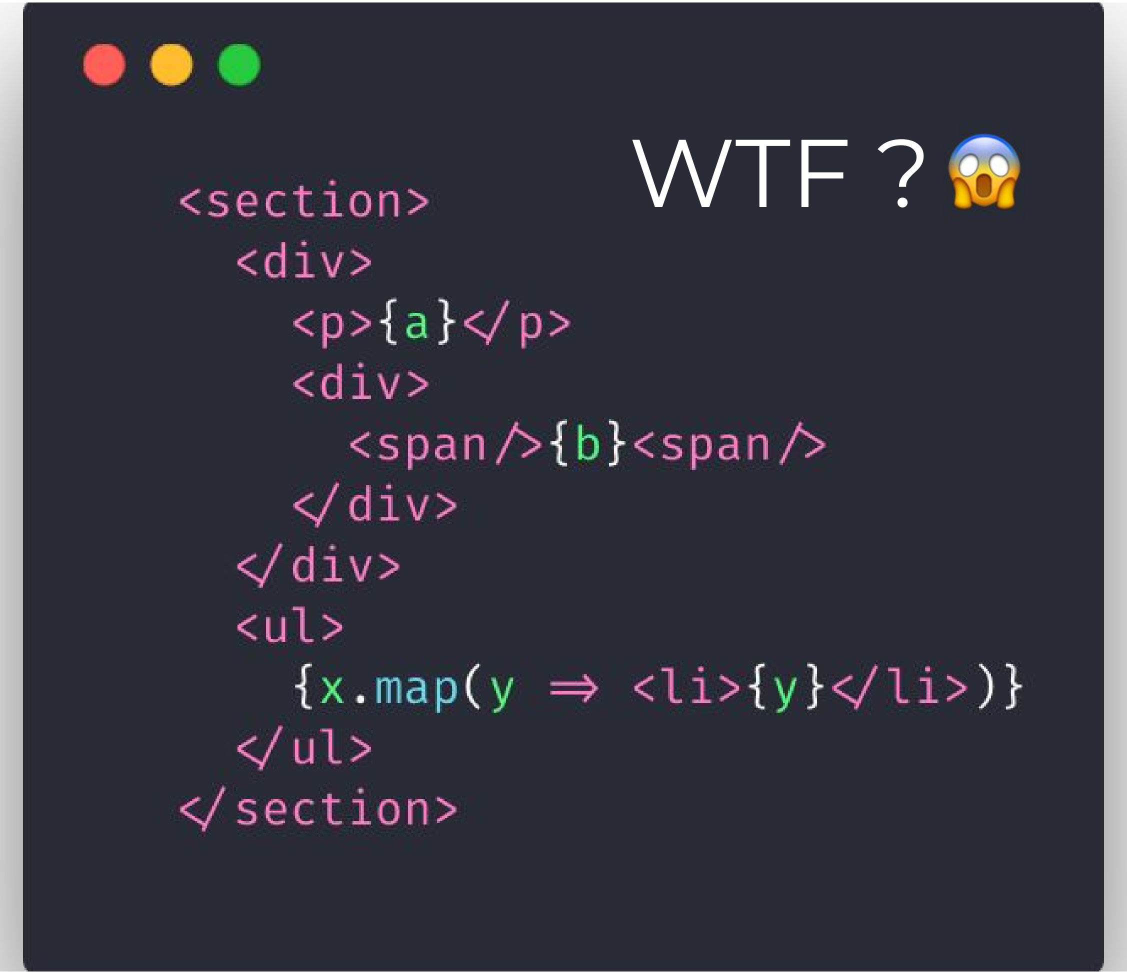
What’s the problem!?

@zealigan



```
<section>
  <div>
    <p>{a}</p>
    <div>
      <span/>{b}<span/>
    </div>
  </div>
  <ul>
    {x.map(y => <li>{y}</li>)}
  </ul>
</section>
```

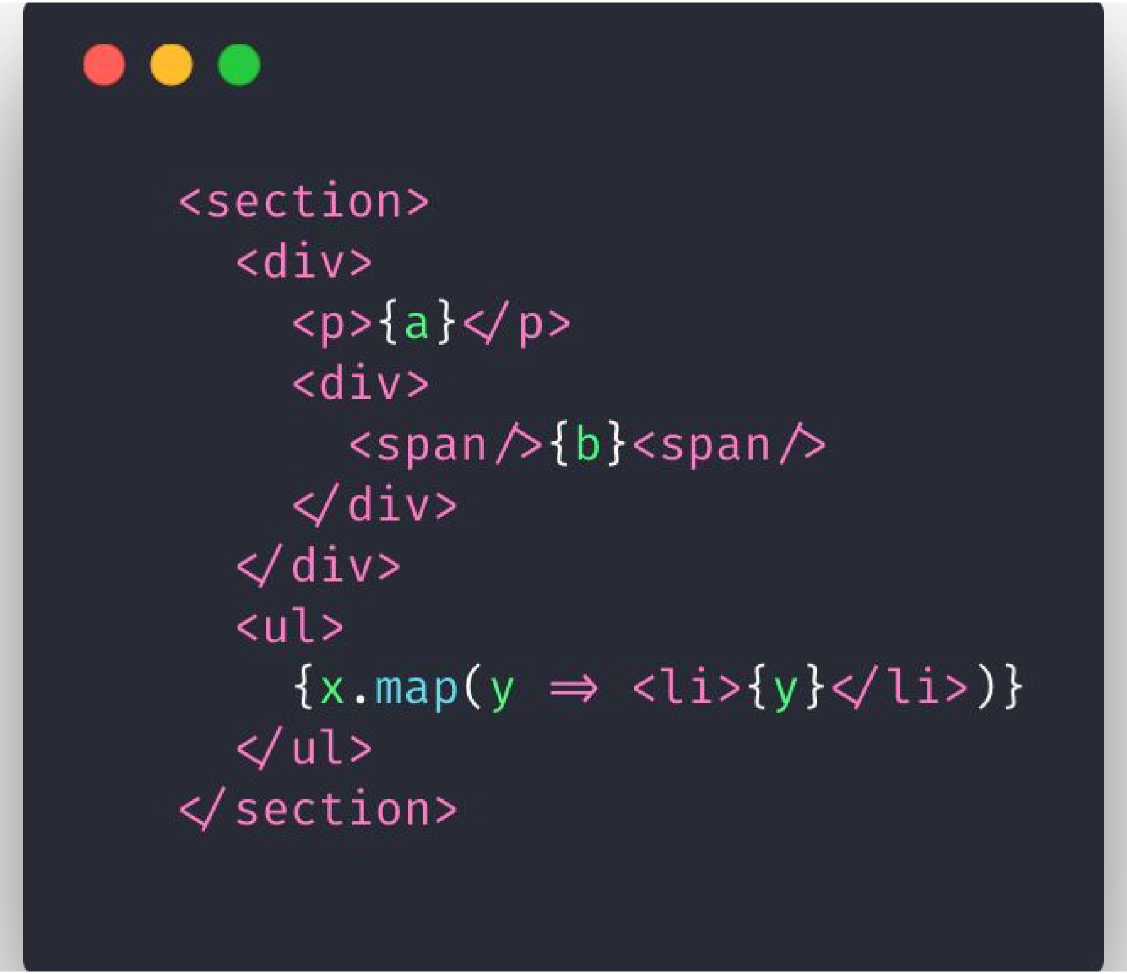
@zealigan



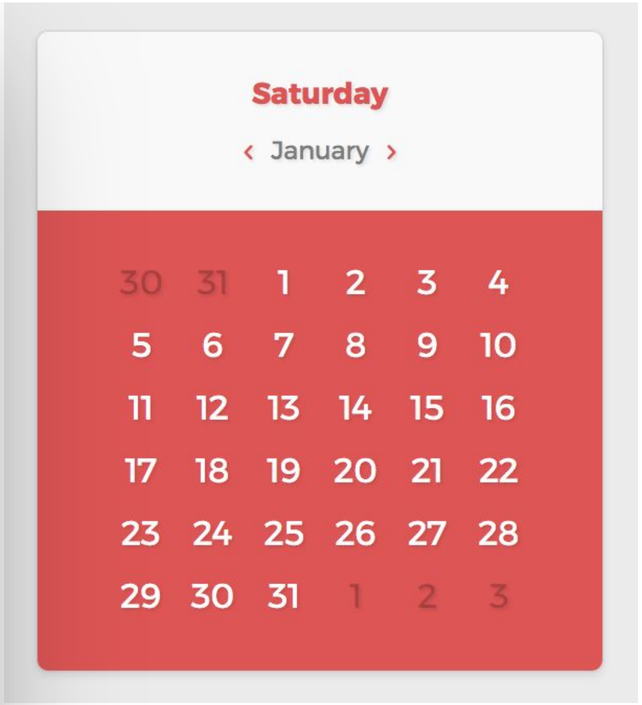
WTF ? 😱

```
<section>
  <div>
    <p>{a}</p>
    <div>
      <span/>{b}<span/>
    </div>
  </div>
  <ul>
    {x.map(y => <li>{y}</li>)}
  </ul>
</section>
```

@zealigan



```
<section>
  <div>
    <p>{a}</p>
    <div>
      <span/>{b}<span/>
    </div>
  </div>
  <ul>
    {x.map(y => <li>{y}</li>)}
  </ul>
</section>
```



@zealigan

IT'S A … CALENDAR ? 🤷‍♂️

HTML is naturally imperative and relies on semantics to achieve readability

What’s the solution?

Declarative Programming

In computer science, **declarative programming** is a programming paradigm—a style of building the structure and elements of computer programs—that expresses the logic of a computation without describing its control flow.[1]

STOP MAKING ASSUMPTIONS

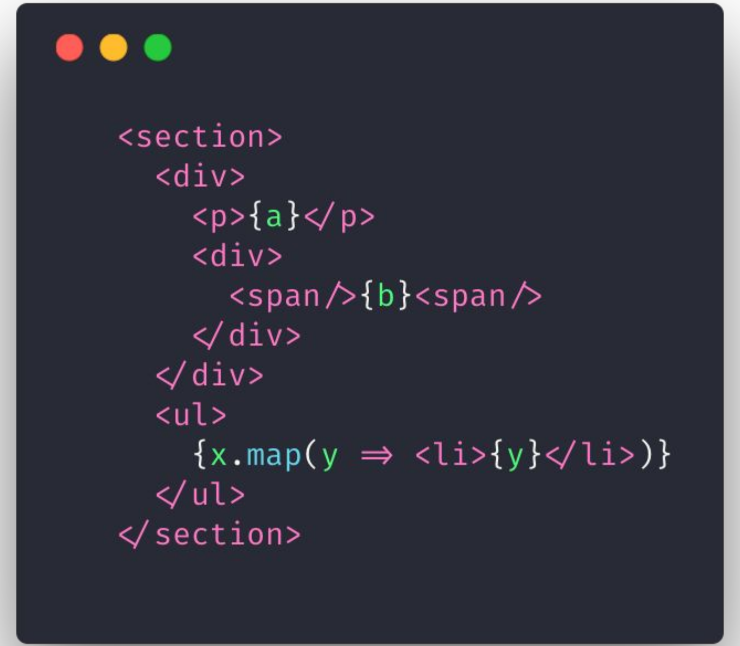
Let’s change the display of the days of the calendar.

Let’s change the display of the days of the calendar.

Easy … right?

Let’s change the display of the days of the calendar.

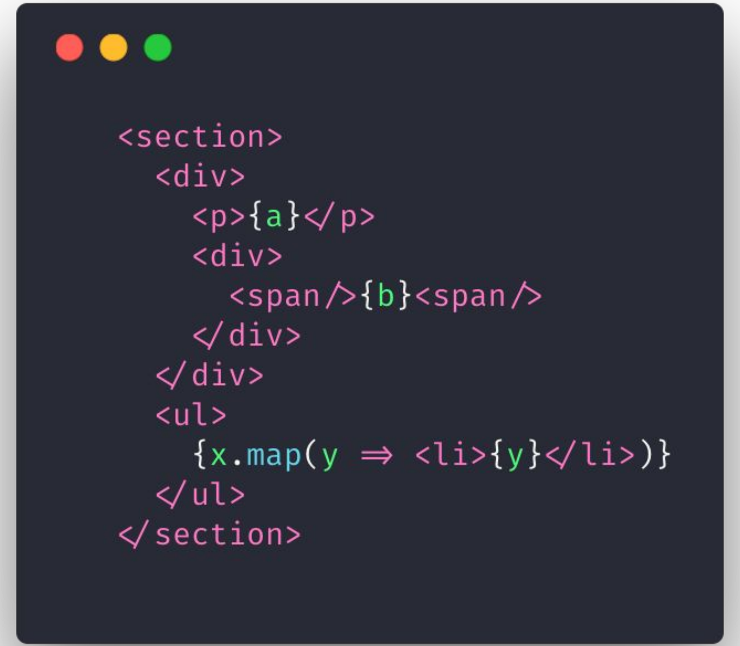
WHERE ARE THE DAYS?



```
<section>
  <div>
    <p>{a}</p>
    <div>
      <span/>{b}<span/>
    </div>
  </div>
  <ul>
    {x.map(y => <li>{y}</li>)}
  </ul>
</section>
```

Let’s change the display of the days of the calendar.

You are guessing what element the days are.



```
<section>
  <div>
    <p>{a}</p>
    <div>
      <span/>{b}<span/>
    </div>
  </div>
  <ul>
    {x.map(y => <li>{y}</li>)}
  </ul>
</section>
```

@zealigan

What about CSS?

Doesn’t CSS solve the problem?!

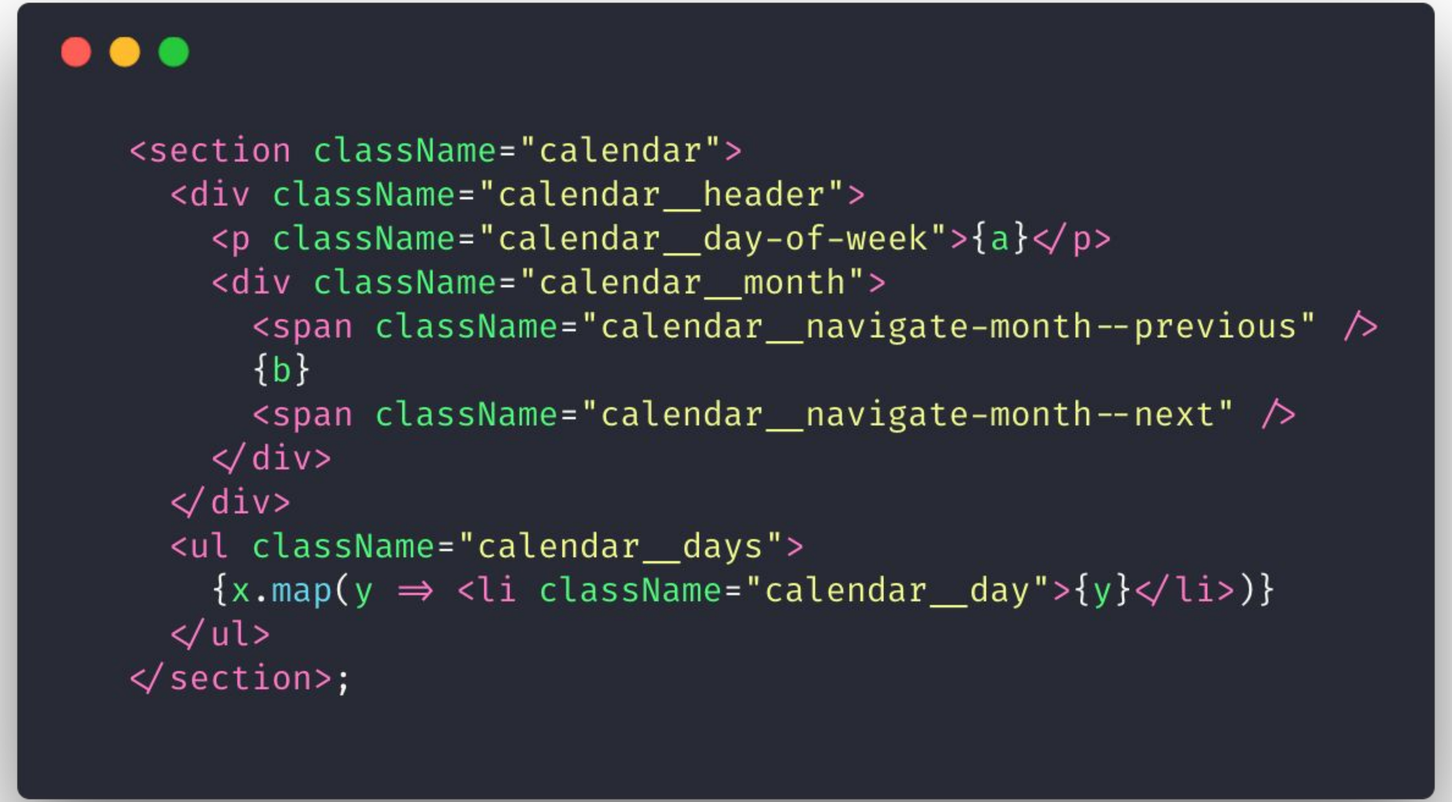
What about CSS?

Doesn't CSS solve the problem?!

- Yup

What about CSS?

Using conventions like BEM



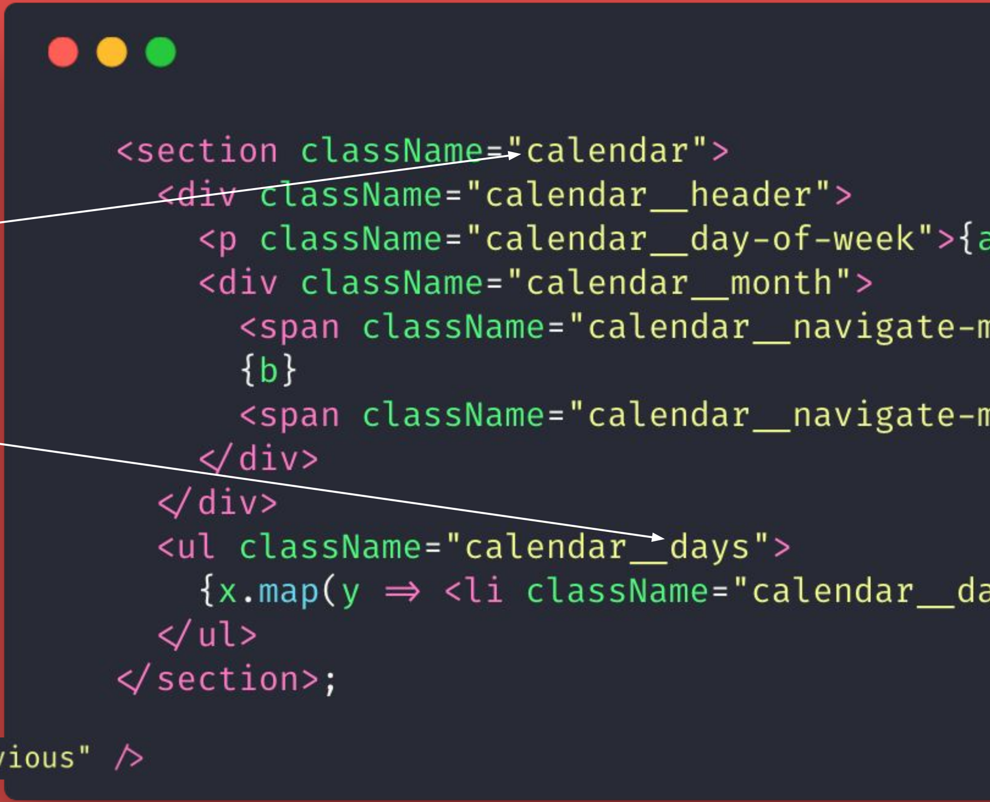
```
<section className="calendar">
  <div className="calendar__header">
    <p className="calendar__day-of-week">{a}</p>
    <div className="calendar__month">
      <span className="calendar__navigate-month--previous" />
      {b}
      <span className="calendar__navigate-month--next" />
    </div>
  </div>
  <ul className="calendar__days">
    {x.map(y => <li className="calendar__day">{y}</li>)}
  </ul>
</section>;
```

@zealigan

Block

Element

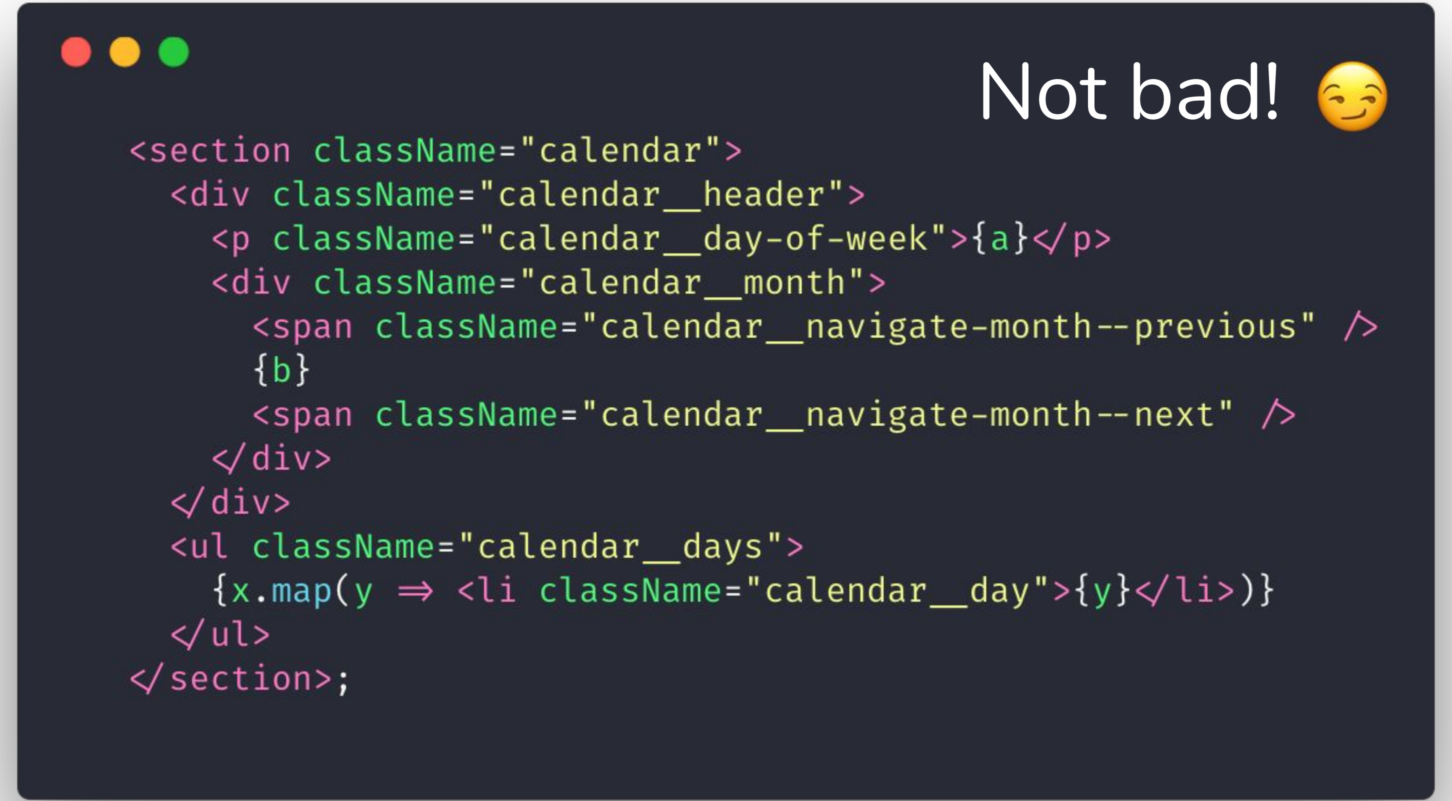
Modifier



```
<section className="calendar">
  <div className="calendar__header">
    <p className="calendar__day-of-week">{a
    <div className="calendar__month">
      <span className="calendar__navigate-m
      {b}
      <span className="calendar__navigate-m
    </div>
  </div>
  <ul className="calendar__days">
    {x.map(y => <li className="calendar__da
  </ul>
</section>;
```

```
<span className="calendar__navigate-month--previous" />
```

@zealigan



Not bad! 😏

```
<section className="calendar">
  <div className="calendar__header">
    <p className="calendar__day-of-week">{a}</p>
    <div className="calendar__month">
      <span className="calendar__navigate-month--previous" />
      {b}
      <span className="calendar__navigate-month--next" />
    </div>
  </div>
  <ul className="calendar__days">
    {x.map(y => <li className="calendar__day">{y}</li>)}
  </ul>
</section>;
```

@zealigan

What about CSS?

There are a few new problems we have introduced

Problems with CSS

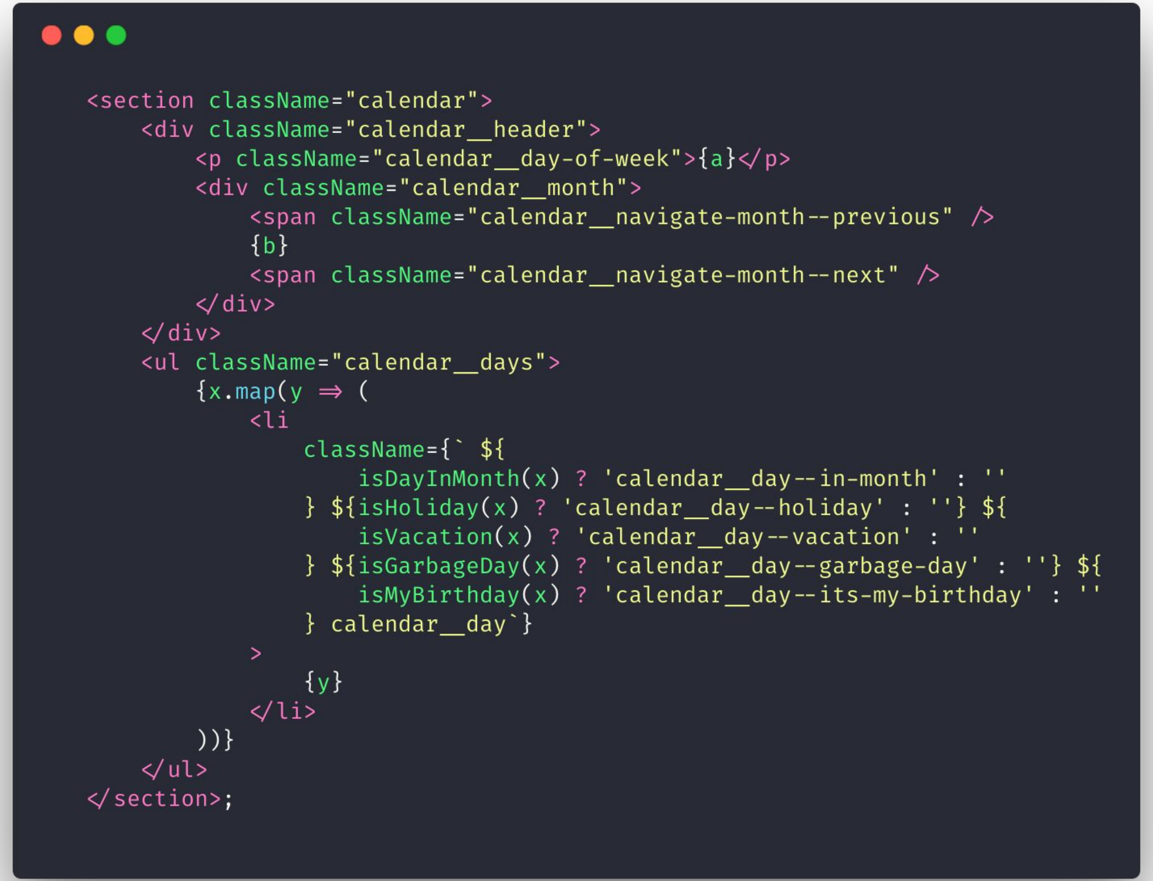
- Namespacing
- Webpack config
- Conventions
- Conditional Classes 😭

@zealigan

Let’s change the display of the days of the calendar, with BEM.

```jsx
<section className="calendar">
    <div className="calendar__header">
        <p className="calendar__day-of-week">{a}</p>
        <div className="calendar__month">
            <span className="calendar__navigate-month--previous" />
            {b}
            <span className="calendar__navigate-month--next" />
        </div>
    </div>
    <ul className="calendar__days">
        {x.map(y => (
            <li
                className={`${
                    isDayInMonth(x) ? 'calendar__day--in-month' : ''
                } calendar__day`}
            >
                {y}
            </li>
        ))}
    </ul>
</section>;
```

It works. We have more context. But if we have more than one conditional class …



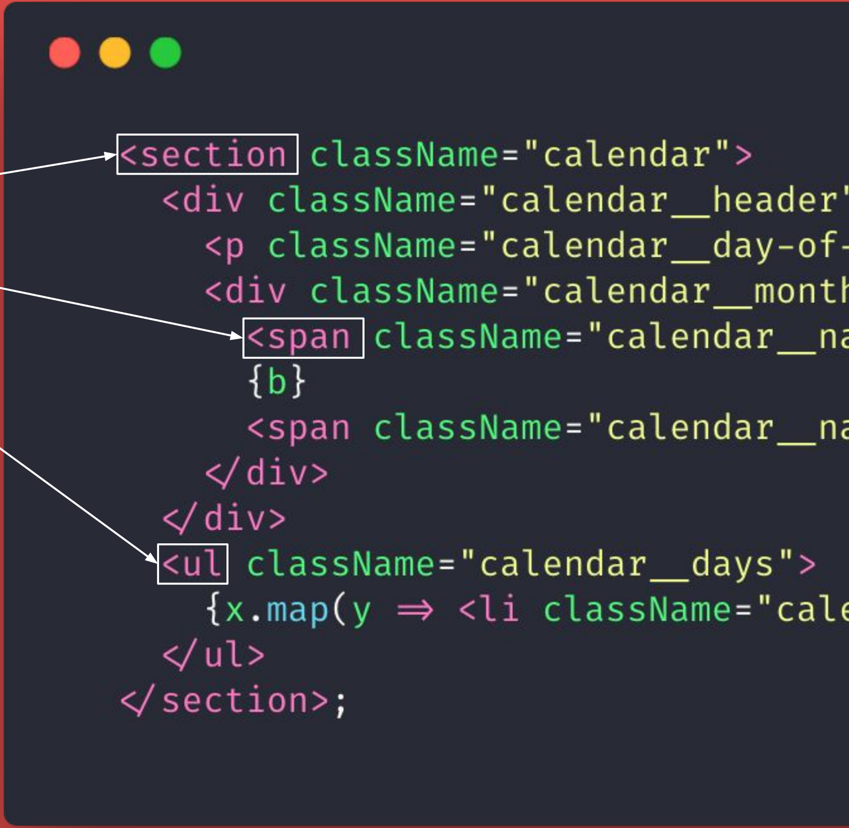
```
<section className="calendar">
    <div className="calendar__header">
        <p className="calendar__day-of-week">{a}</p>
        <div className="calendar__month">
            <span className="calendar__navigate-month--previous" />
            {b}
            <span className="calendar__navigate-month--next" />
        </div>
    </div>
    <ul className="calendar__days">
        {x.map(y => (
            <li
                className={` ${
                    isDayInMonth(x) ? 'calendar__day--in-month' : ''
                } ${isHoliday(x) ? 'calendar__day--holiday' : ''} ${
                    isVacation(x) ? 'calendar__day--vacation' : ''
                } ${isGarbageDay(x) ? 'calendar__day--garbage-day' : ''} ${
                    isMyBirthday(x) ? 'calendar__day--its-my-birthday' : ''
                } calendar__day`}
            >
                {y}
            </li>
        ))}
    </ul>
</section>;
```

@zealigan

Is it really declarative?

These don't mean anything to us as designers and developers

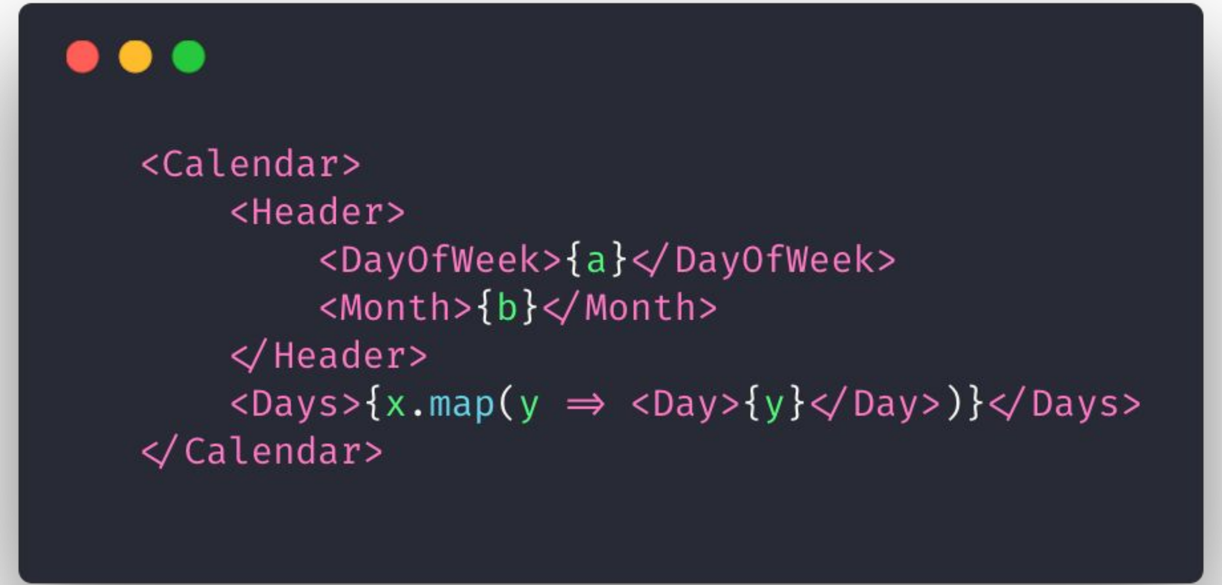
They are <u>contextless</u>



```
<section className="calendar">
  <div className="calendar__header'
    <p className="calendar__day-of-
    <div className="calendar__month
      <span className="calendar__na
      {b}
      <span className="calendar__na
    </div>
  </div>
  <ul className="calendar__days">
    {x.map(y => <li className="cale
  </ul>
</section>;
```

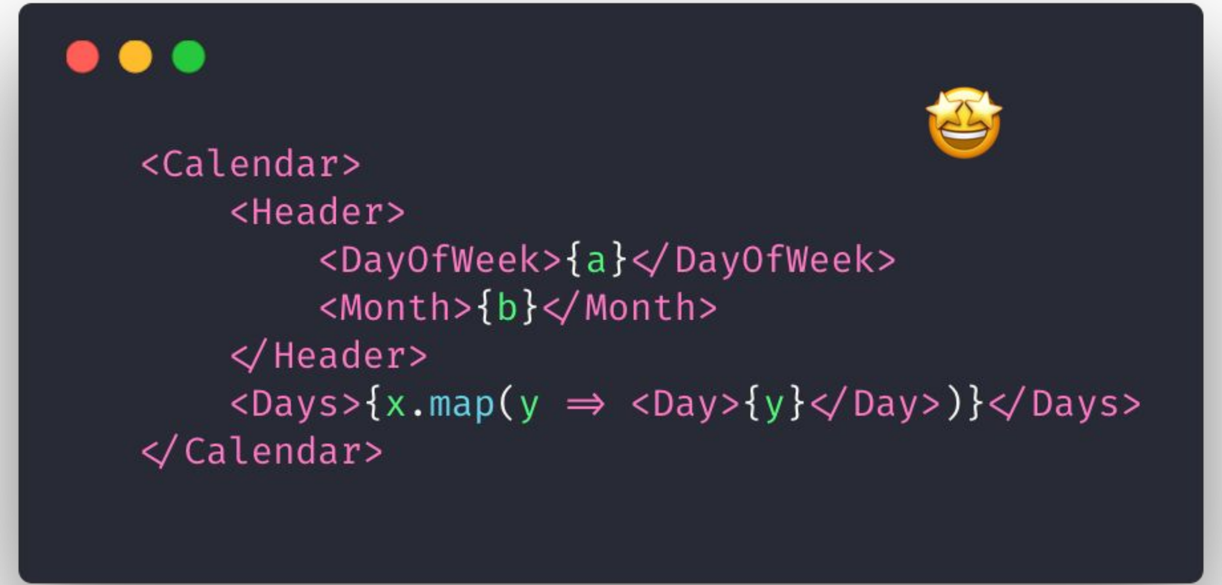
@zealigan

So what does declarative really look like?



```
<Calendar>
    <Header>
        <DayOfWeek>{a}</DayOfWeek>
        <Month>{b}</Month>
    </Header>
    <Days>{x.map(y => <Day>{y}</Day>)}</Days>
</Calendar>
```

@zealigan



```
<Calendar>
    <Header>
        <DayOfWeek>{a}</DayOfWeek>
        <Month>{b}</Month>
    </Header>
    <Days>{x.map(y => <Day>{y}</Day>)}</Days>
</Calendar>
```

@zealigan

BUT THAT IS JUST STYLED COMPONENTS

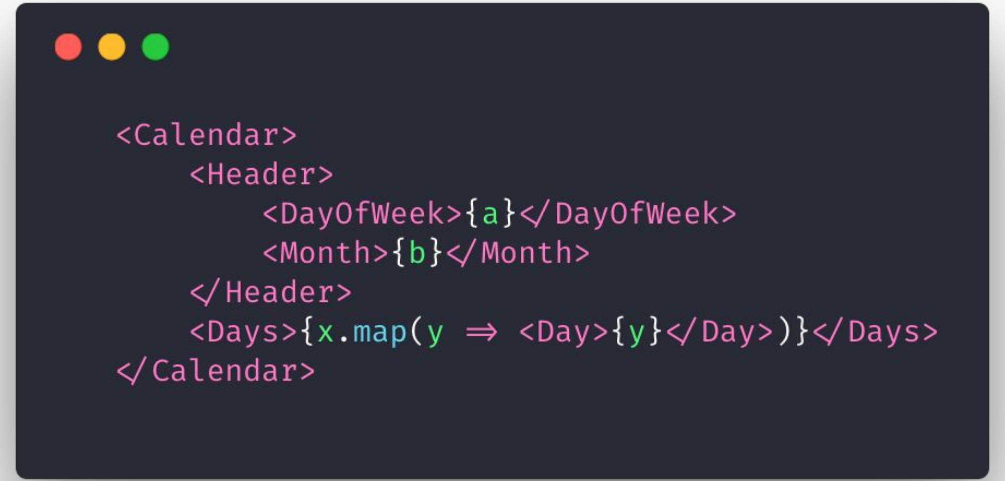
That style of coding was a HUGE win for our team.

Now people can make independent, informed changes!

Let’s change the display of the days of the calendar, with Styled-Components.

Let’s change the display of the days of the calendar, with Styled-Components.

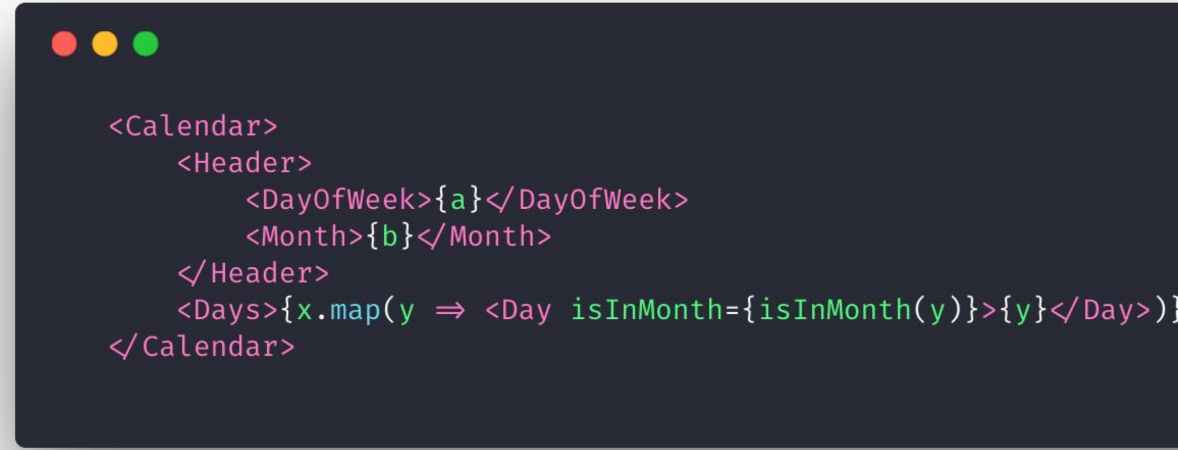
We know exactly the component to change!



```
<Calendar>
    <Header>
        <DayOfWeek>{a}</DayOfWeek>
        <Month>{b}</Month>
    </Header>
    <Days>{x.map(y => <Day>{y}</Day>)}</Days>
</Calendar>
```

Let’s change the display of the days of the calendar, with Styled-Components.

We can declaratively add conditionals as props!

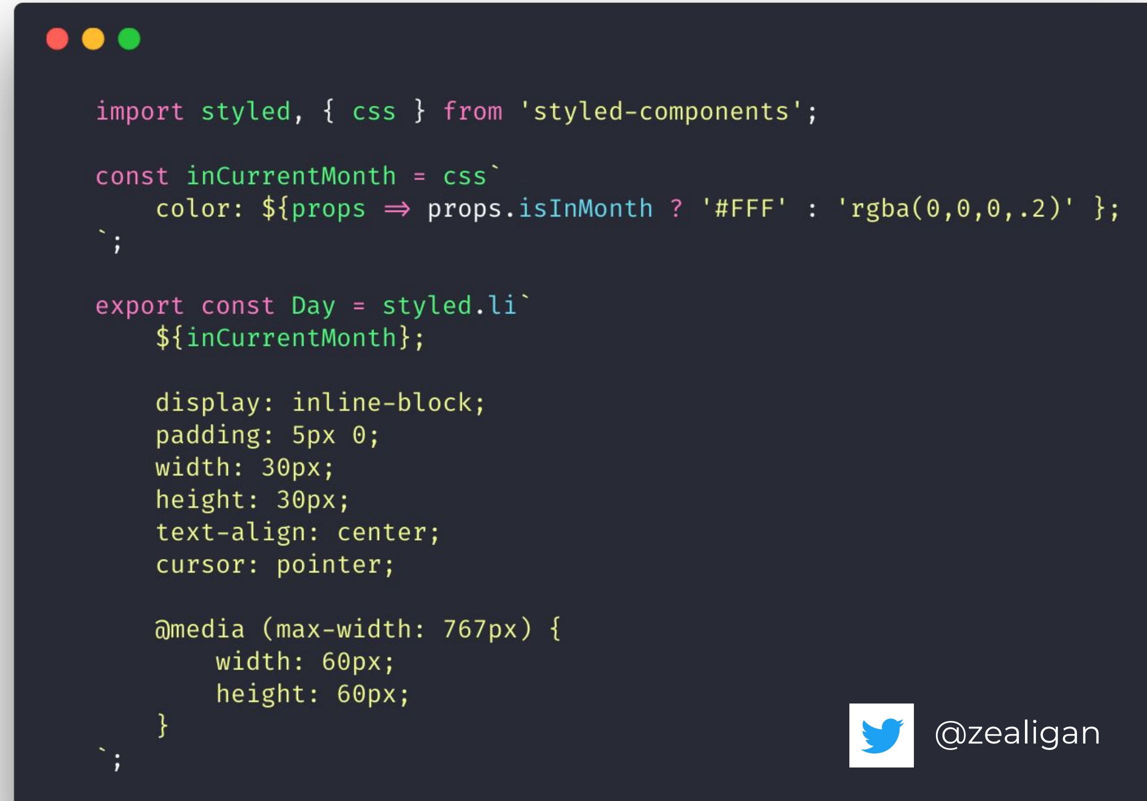


```
<Calendar>
    <Header>
        <DayOfWeek>{a}</DayOfWeek>
        <Month>{b}</Month>
    </Header>
    <Days>{x.map(y => <Day isInMonth={isInMonth(y)}>{y}</Day>)
</Calendar>
```

@zealigan

Let’s change the display of the days of the calendar, with Styled-Components.

In the styled component

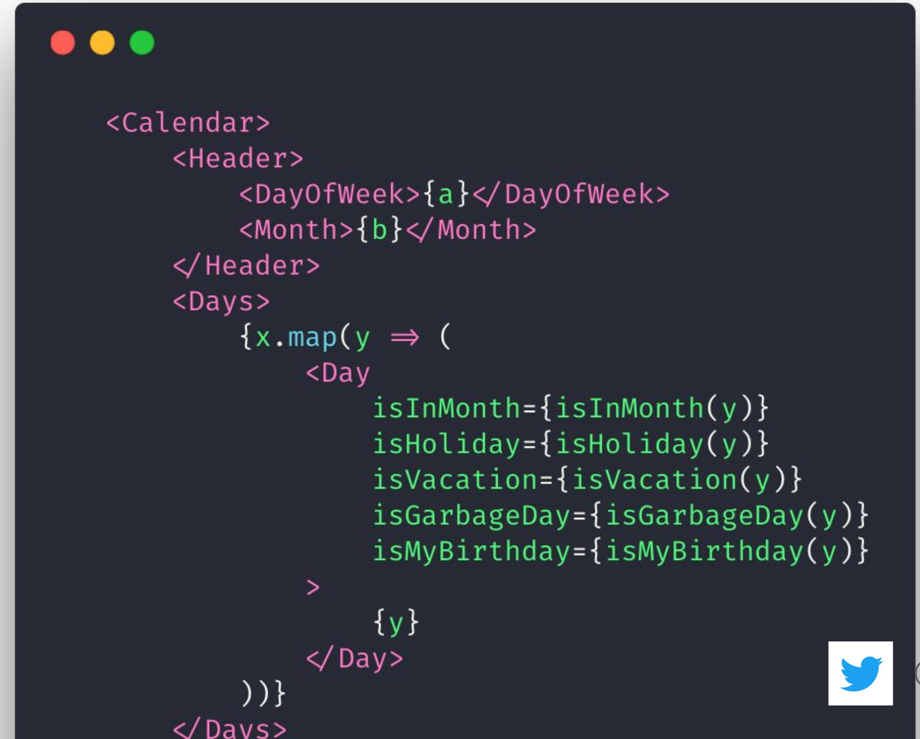


```
import styled, { css } from 'styled-components';

const inCurrentMonth = css`
    color: ${props ⇒ props.isInMonth ? '#FFF' : 'rgba(0,0,0,.2)' };
`;

export const Day = styled.li`
    ${inCurrentMonth};

    display: inline-block;
    padding: 5px 0;
    width: 30px;
    height: 30px;
    text-align: center;
    cursor: pointer;

    @media (max-width: 767px) {
        width: 60px;
        height: 60px;
    }
`;
```

@zealigan

Let’s change the display of the days of the calendar, with Styled-Components.

We can even add more without clouding the code!



```
<Calendar>
    <Header>
        <DayOfWeek>{a}</DayOfWeek>
        <Month>{b}</Month>
    </Header>
    <Days>
        {x.map(y => (
            <Day
                isInMonth={isInMonth(y)}
                isHoliday={isHoliday(y)}
                isVacation={isVacation(y)}
                isGarbageDay={isGarbageDay(y)}
                isMyBirthday={isMyBirthday(y)}
            >
                {y}
            </Day>
        ))}
    </Days>
```

@zealigan

Styled Components 💅

CSS is actually pretty good!

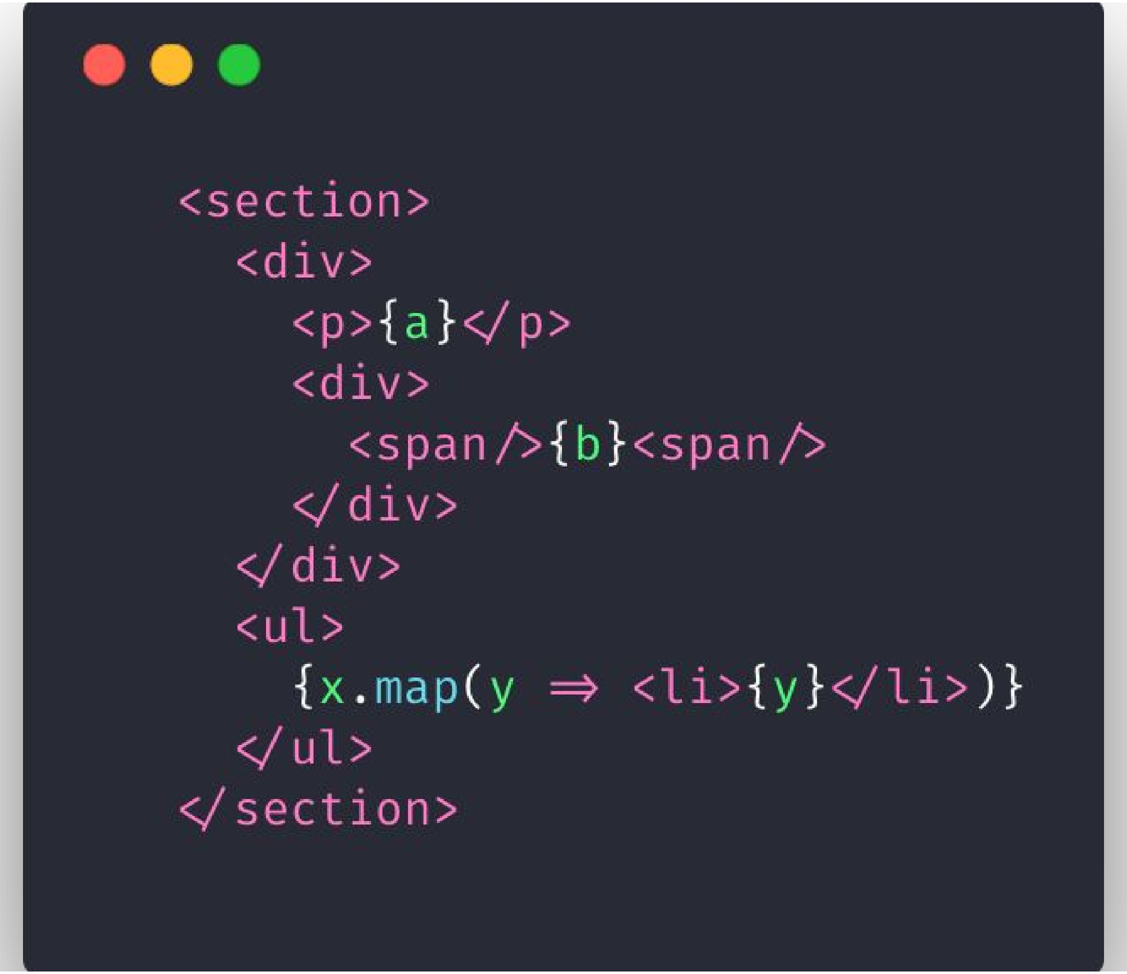
We should keep writing it.

Styled Components makes that easy.

Styled Components 💅

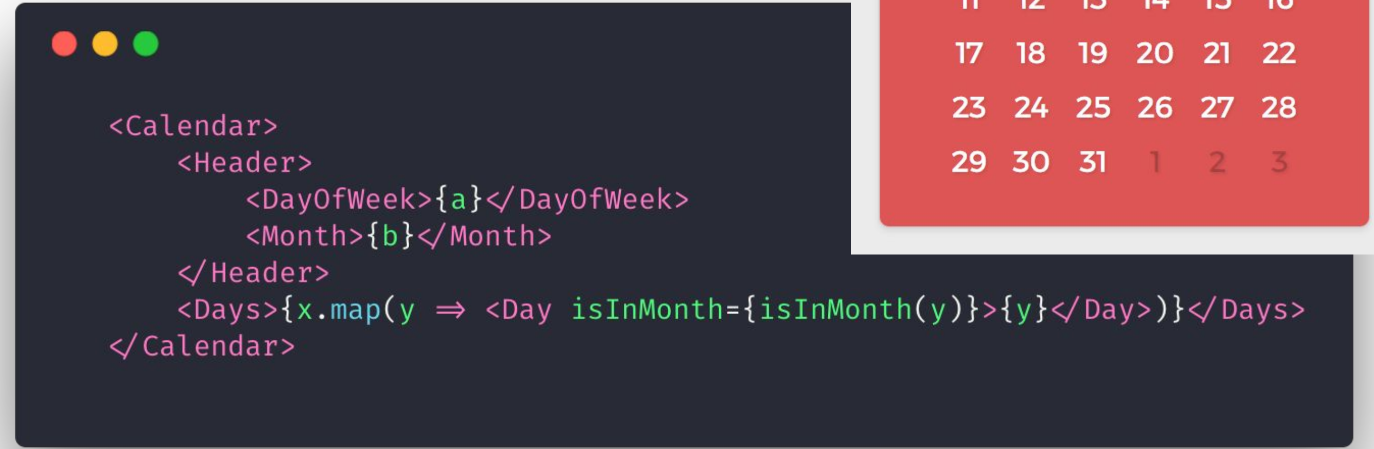
Plus it gives us:

- Auto CSS prefixing
- Only sending CSS to the browser for components that are on the page!
- You can even use them in React Native!

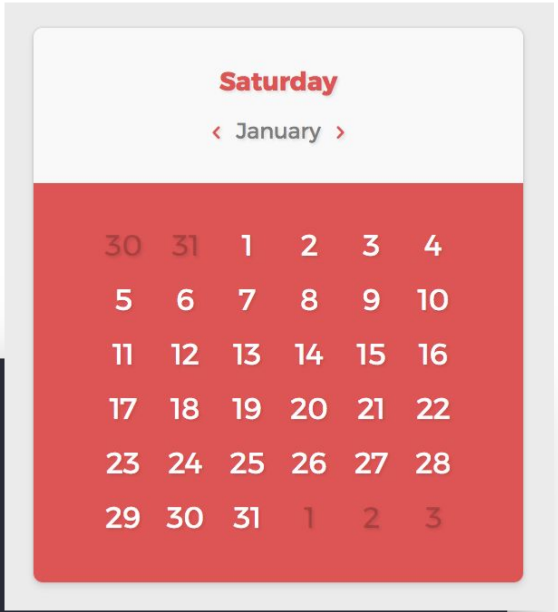


```
<section>
  <div>
    <p>{a}</p>
    <div>
      <span/>{b}<span/>
    </div>
  </div>
  <ul>
    {x.map(y => <li>{y}</li>)}
  </ul>
</section>
```

@zealigan



```
<Calendar>
    <Header>
        <DayOfWeek>{a}</DayOfWeek>
        <Month>{b}</Month>
    </Header>
    <Days>{x.map(y => <Day isInMonth={isInMonth(y)}>{y}</Day>)}</Days>
</Calendar>
```



@zealigan

🙇 Thank you 👏

Source:
https://github.com/mb3online/YOW-styled-components

@zealigan