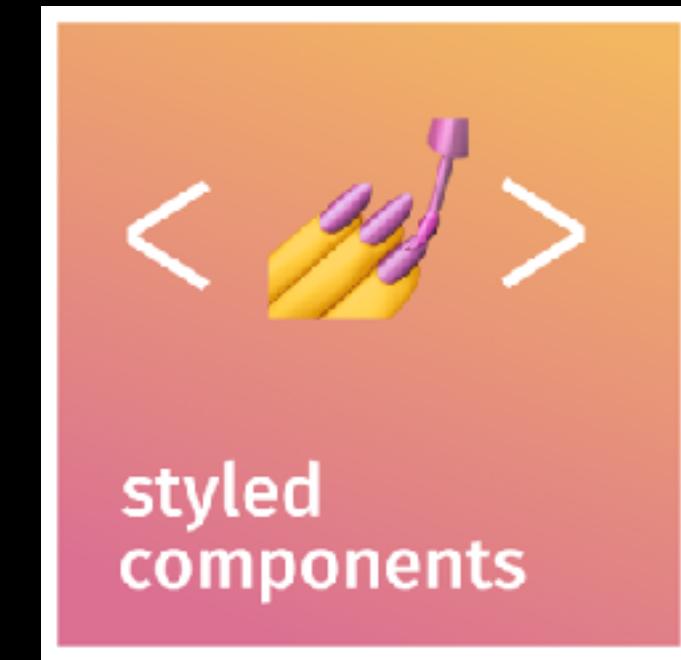


Hei
Finland





styled-components

more complicated stuff?



Max Stoiber
@mxstbr



Glen Maddern
@glenmaddern

Max Stoiber - creator of Styled-Components

Glen Maddern - creator of CSS-Modules



you are the {css} to my <html>



you are styled-components
to my react



Kasia Jastrzębska
@kejt_bw



a DAMCO innovation

Styled-components are dope

BUT

...before we go there...

Do I need css-in-js
to have a successful project?

HELL NO!

There is plenty of options
to handle CSS

- ★ good old CSS
- ★ CSS processors as postCSS, Sass, Less
- ★ using methodologies like BEM
- ★ CSS Modules



then why even bother?

Because of the advantages!

- ★ Components

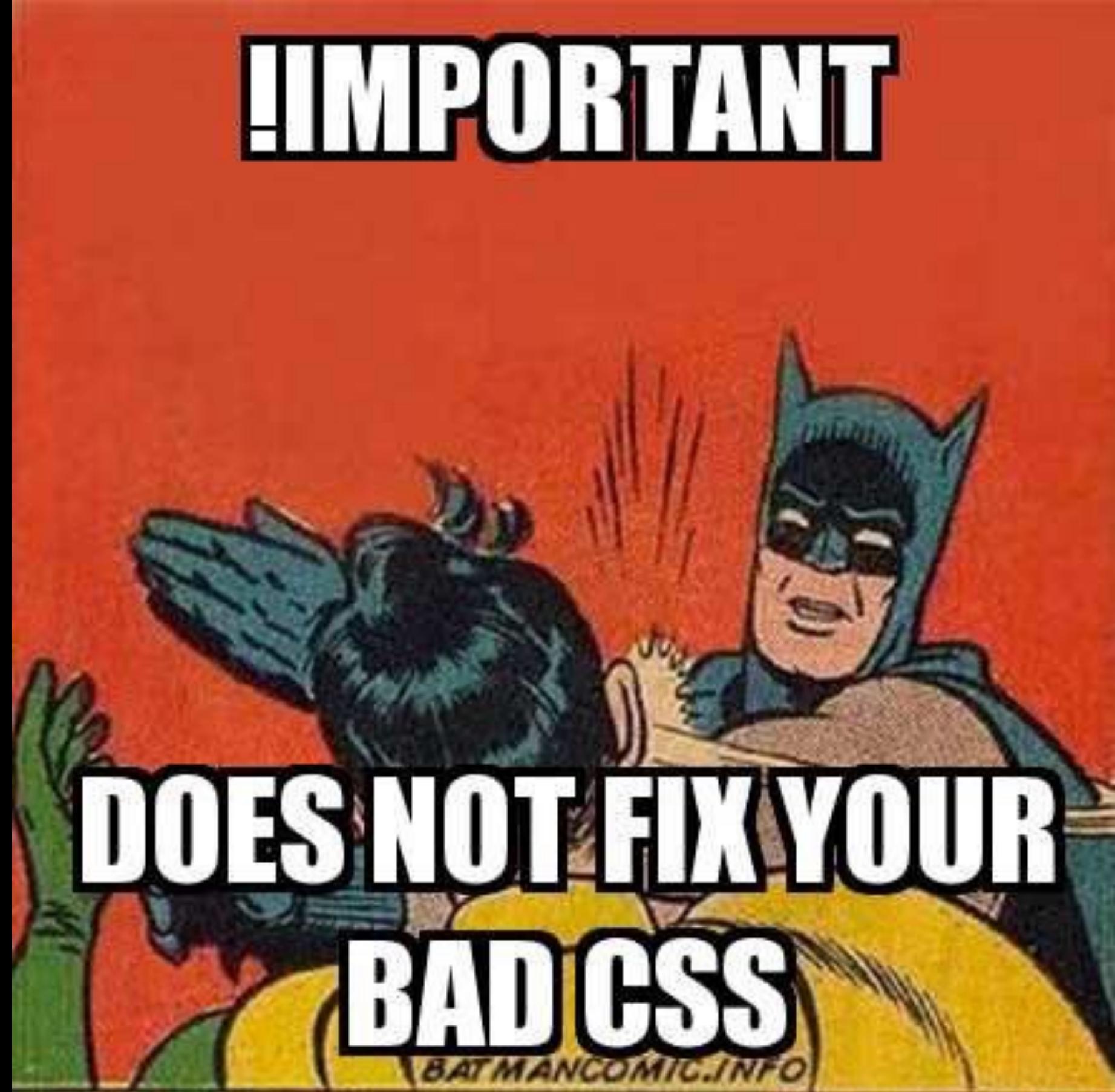
- ★ Power and simplicity of CSS
- ★ Power of Javascript
- ★ No need to worry about overwriting
- ★ SSR included
- ★ Theming made easy!

DISCLAIMER!

YOU STILL HAVE TO KNOW CSS

...no more !important
in your codebase

!IMPORTANT



**DOES NOT FIX YOUR
BAD CSS**

BATMANCOMIC.INFO

there are few css related
concerns

global vs local

```
/* Component needs to be pink */  
* {  
    color: pink;  
    background: pink;  
}
```

by default in CSS everything is global

even if you wanted to color only
one thing



```
import React from 'react';
import styled from 'styled-components';

export default styled.h1`  
  font-size: 1.5em;  
`
```

Because javascript

in styled-components everything is local by default

```
import Title from './components/Title'

class App extends Component {
  render() {
    return (
      <Wrapper>
        <Title>Welcome to Styled React</Title>
      </Wrapper>
    );
  }
}
```

in styled-components everything is local by default

```
<style data-styled-components="">
  /* sc-component-id: sc-bdVaJa */
  .dAIgSr{font-size:1.5em;}
  /* sc-component-id: sc-bwzfXH */
  .gJIwTT{text-align:center;}
</style>
```

generated css goes to the end of html head

what if I want global styles

```
injectGlobal`  
body {  
  margin: 0;  
  padding: 0;  
  font-family: sans-serif;  
}  
`
```

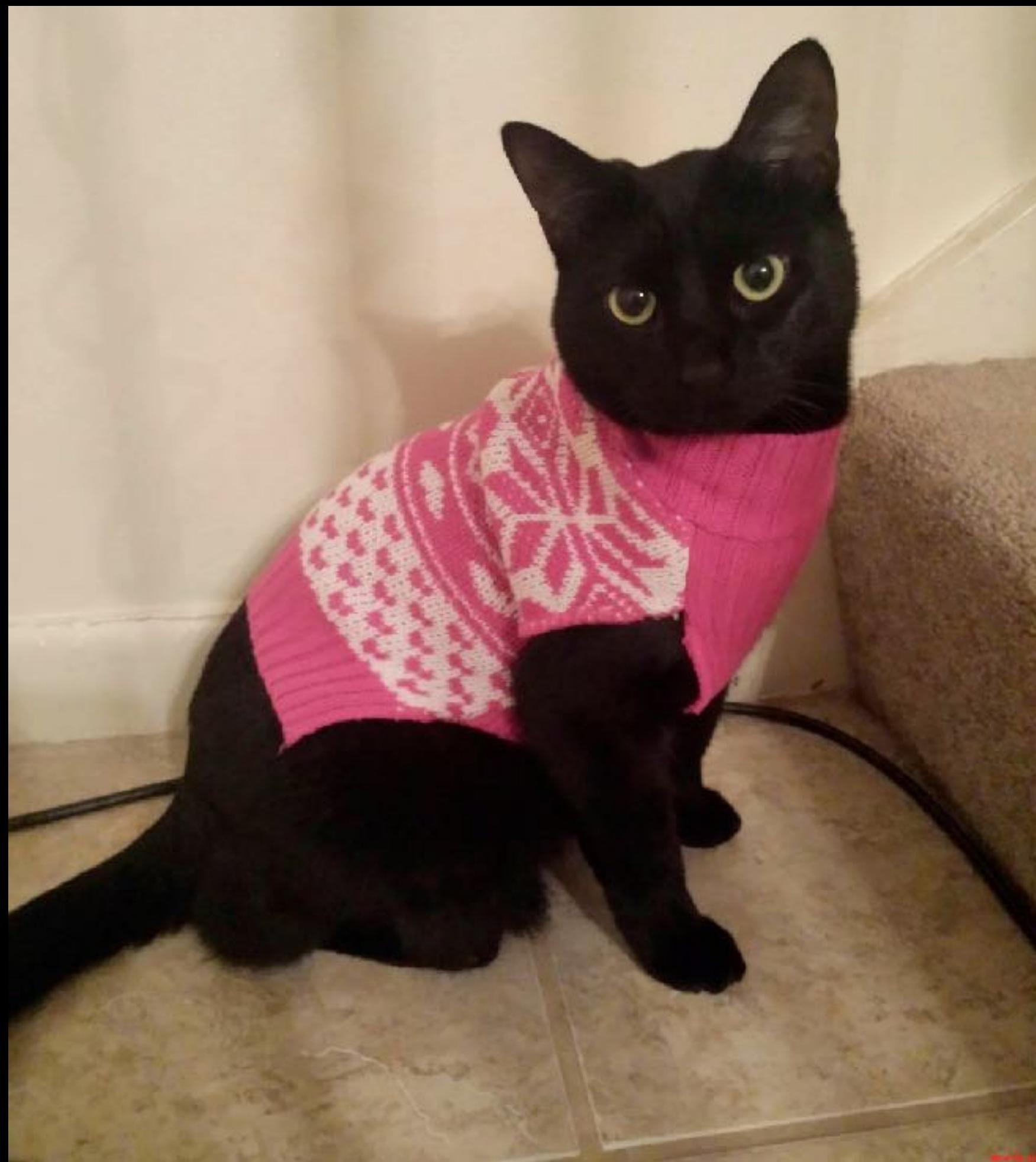
still possible

reusing css code

what drives development?



I want a cat in a sweater







by default
with CSS

```
.underlined {  
    text-decoration: underline;  
}
```

```
.big-sized {  
    font-size: 2em;  
}
```

```
...  
<p className="big-sized underlined">  
    Hello, I am underlined  
</p>  
...
```

with the power of classes

what is the specificity problem?

```
.underlined {  
    text-decoration: underline;  
}  
}
```

```
.big-sized {  
    font-size: 2em;  
}  
}
```

```
p.underlined + p {  
    text-decoration: line-through;  
}  
}
```

```
<p className="big-sized underlined">  
  Hello, I am underlined  
</p>
```

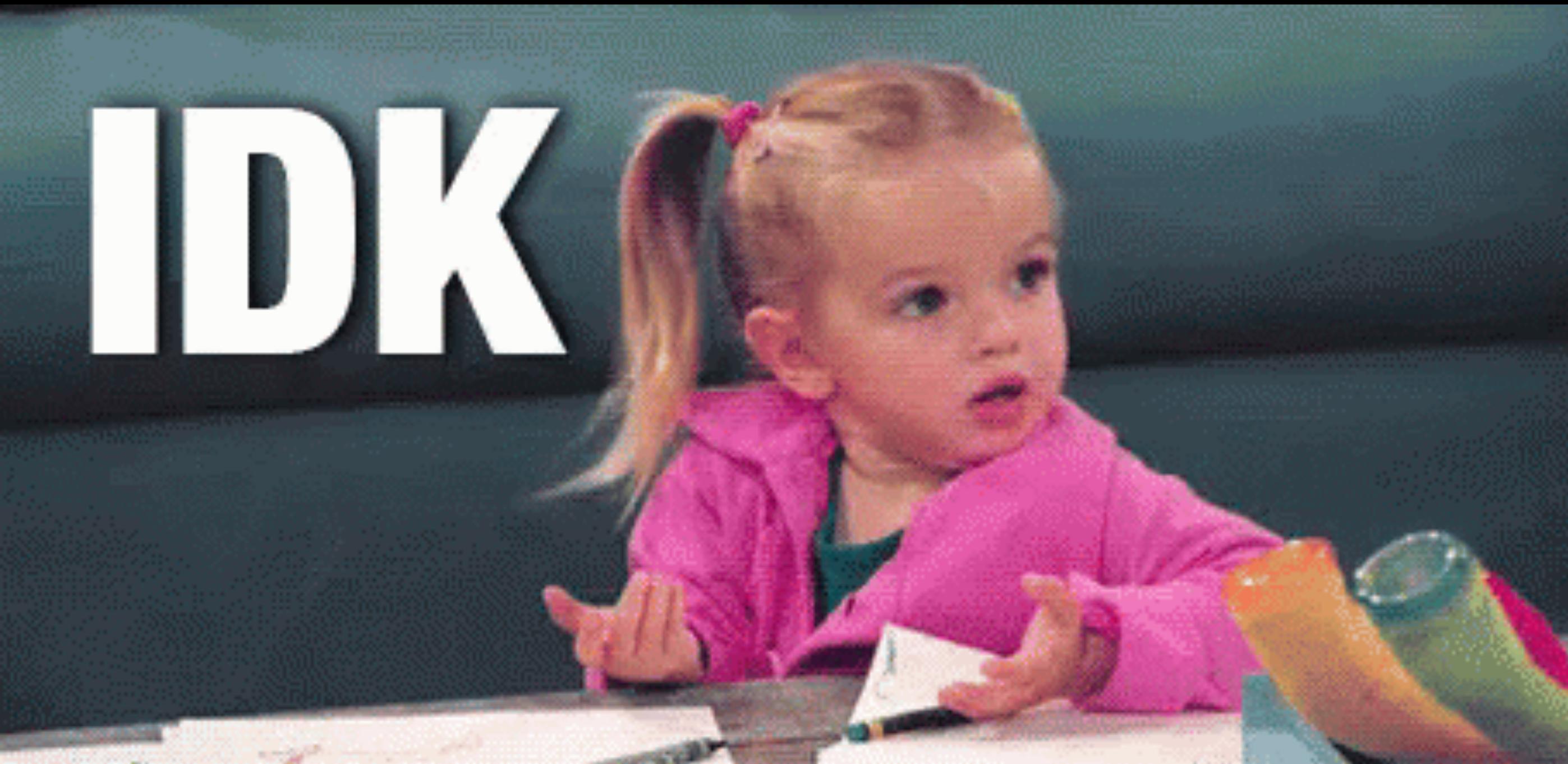
Hello, I am underlined

```
<p className="big-sized underlined">
  Hello, I am underlined
</p>
<p className="big-sized underlined">
  Should I be underlined too?
</p>
```

Hello, I am underlined

Should I be underlined too?

IDK



with styled-components
we reuse CSS as we reuse components

```
import Paragraph from './components/Paragraph'  
const Underlined = styled(Paragraph)`  
  text-decoration: underline;  
`
```

```
<Underlined>  
  Hello, I am underlined  
</Underlined>
```

```
import Paragraph from './components/Paragraph'  
const DivHere = Paragraph.withComponent('div')
```

```
<DivHere>  
  I have Paragraph styles!  
</DivHere>
```

reusing styles with different element

What about weird selectors
and this whole specificity?

shit still can happen

but you are more aware of that

theming app

keeping some style variables

in styled-components...



power of CSS & JS unleashed by React

```
class App extends Component {  
  render() {  
    return (  
      <ThemeProvider theme={themes.buffy}>  
        <Cemetery>  
          <DeathsList />  
        </Cemetery>  
      </ThemeProvider>  
    );  
  }  
}
```

ThemeProvider based on the context API

```
{  
  "buffy": {  
    "baseColor": "#ffffff",  
    "mainColor": "#544b8d",  
    "secondaryColor": "#b7b4db",  
    "palleteExtraColor": "#636aa3",  
    "borderColor": "#636aa3"  
  },  
  ...  
}
```

...applies theme object

```
export default styled.header`  
padding: 20px;  
color: ${props => props.theme.baseColor};  
background: ${props => props.theme.mainColor};  
display: flex;  
align-items: center;  
`
```

which is accessible in every styled component

with this powerful tool
we can think about accessibility

DEMO time!

I am huge fan of Sean Bean



He dies a lot



One does not simply Have a Sean Bean live in a movie

List of the finest Sean Bean deths

Who?

Dr Merrick

Movie?

The Island

How?

Hung by chains and shot through the neck

Who?

006

Movie?

Goldeneye

How?

Crushed by an antenna

Who?

Ned Stark

How?

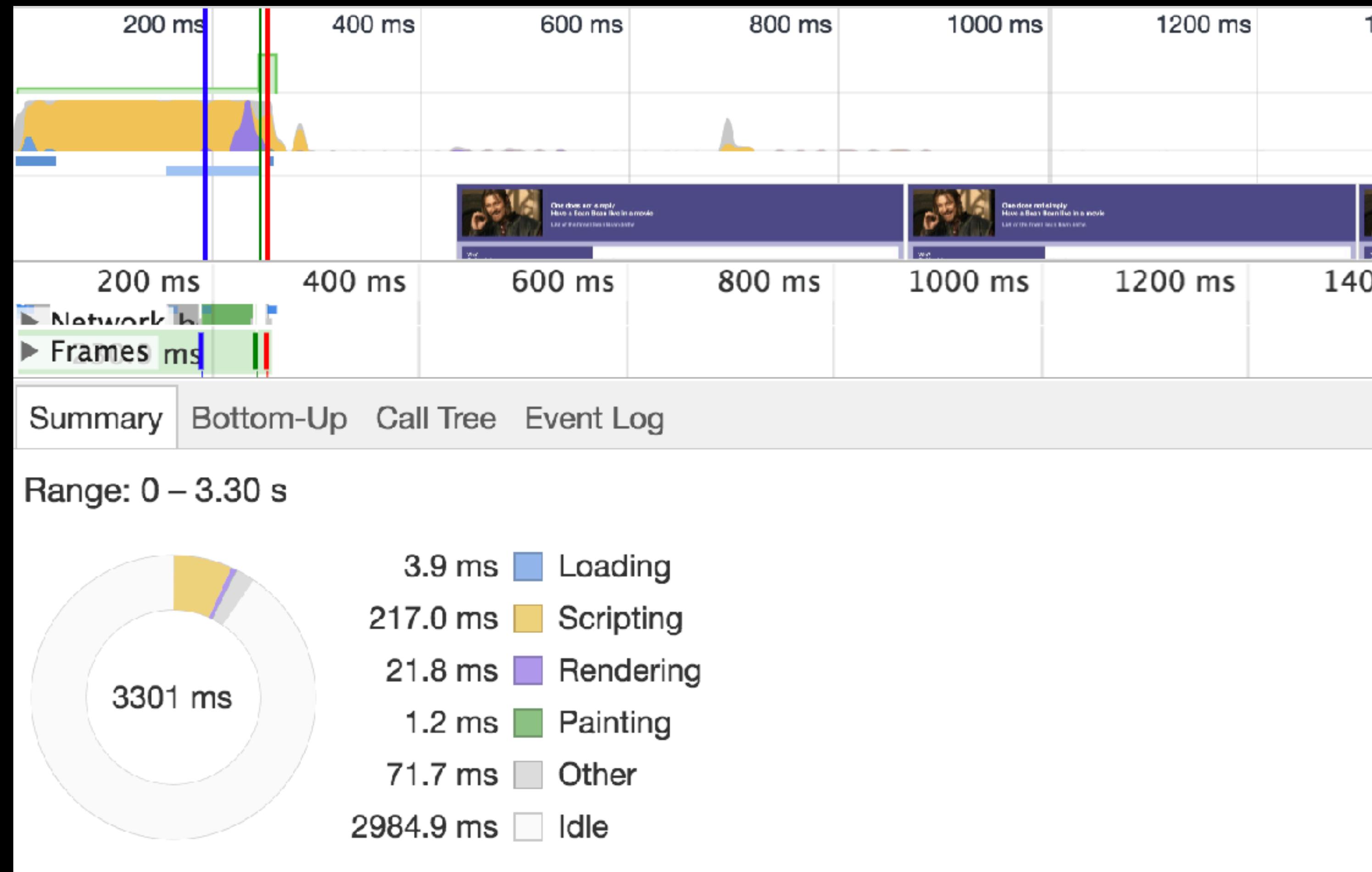
SSR

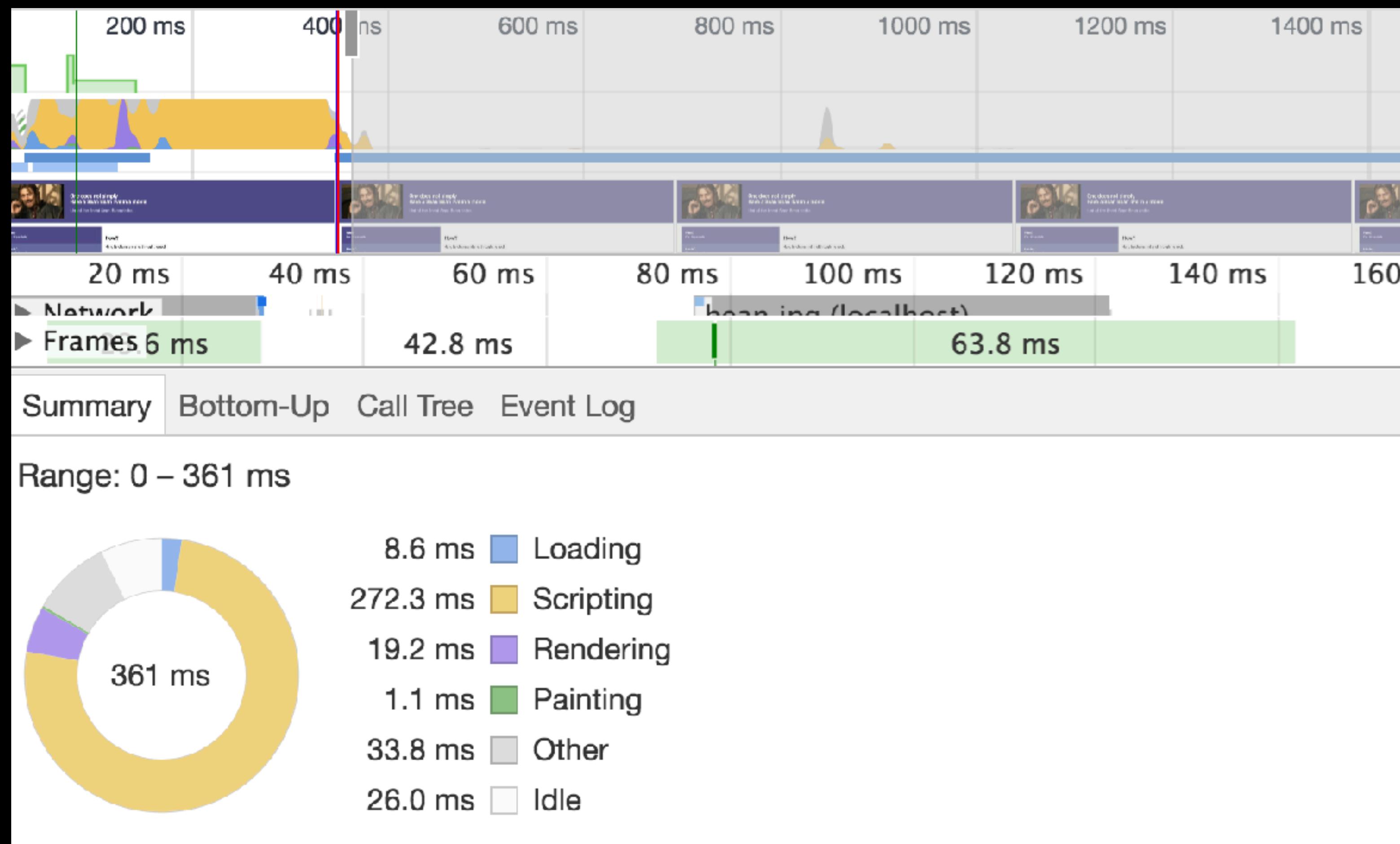
why we should care?

User experience

performance

SEO





how it was done?

```
import { ServerStyleSheet } from 'styled-components'

static getInitialProps ({ renderPage }) {
  const sheet = new ServerStyleSheet()
  const page = renderPage(
    App => props => sheet.collectStyles(<App
      {...props} />))
  const styleTags = sheet.getStyleElement()
  return { ...page, styleTags }
}
```

```
render () {
  return (
    <html>
      <Head>
        <title>My page</title>
        {this.props.styleTags}
      </Head>
      <body>
        <Main />
        <NextScript />
      </body>
    </html>
  )
}
```

TAKE AWAYS

- ★ it's still good old CSS
- ★ helps you organize your code in more modular way
- ★ use theming and JS to improve ACCESSIBILITY
- ★ have fun

every tool is a good tool

thanks!

@kejt_bw