O2
OPEN ORIENTED
凹 凸 实 验 室

CSS-in-JS 介绍

陈俊生

1.什么是 CSS-in-JS？

2.CSS-in-JS 库 emotion

3.总结

什么是 CSS-in-JS ?

“CSS-in-JS 就是用 JS 来写 CSS。”

– unknown

传统 css 痛点

· 全局污染

· 命名混乱

· 样式重用困难

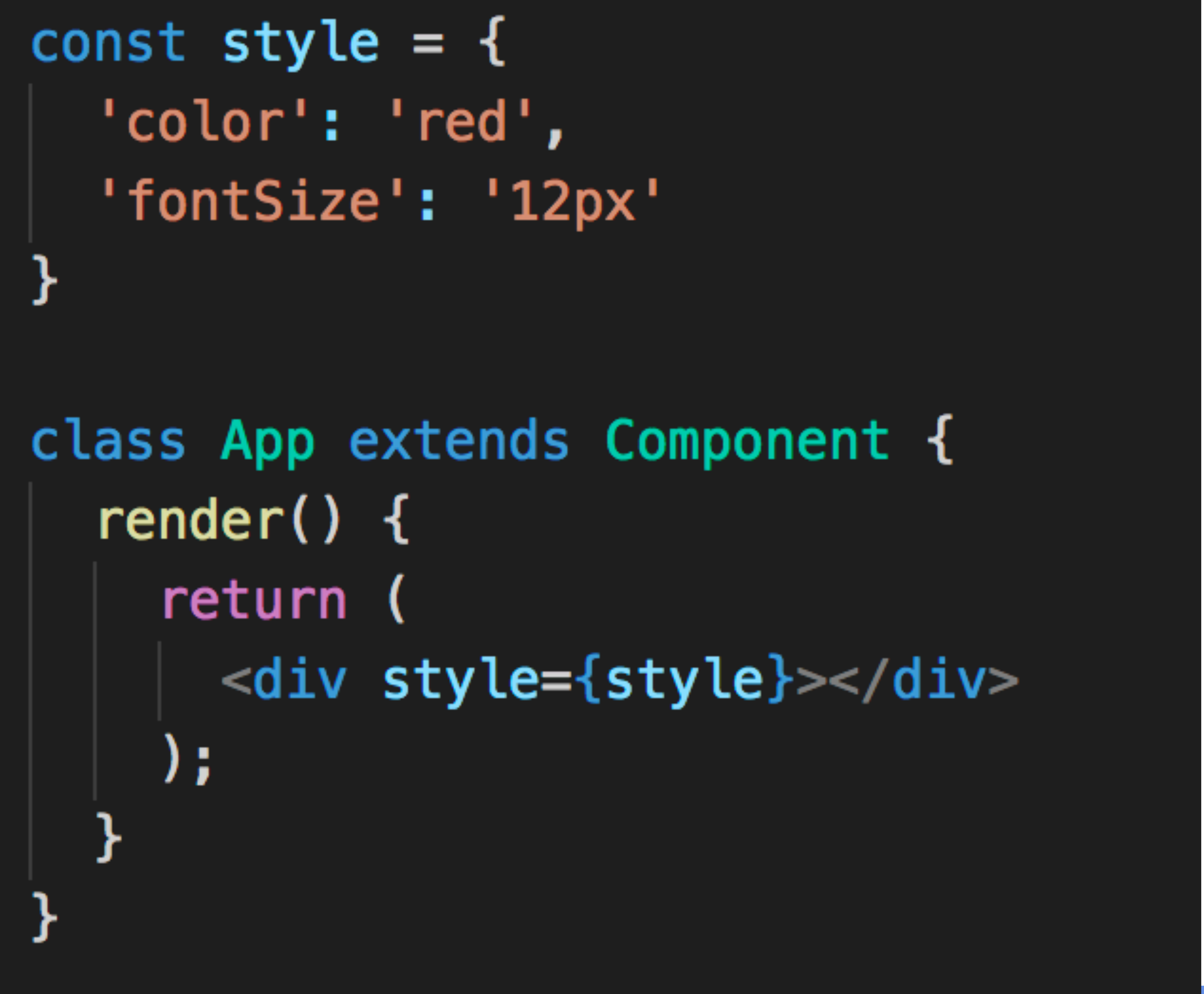
· 代码冗余

· JS 和 CSS 无法共享变量

解决方案

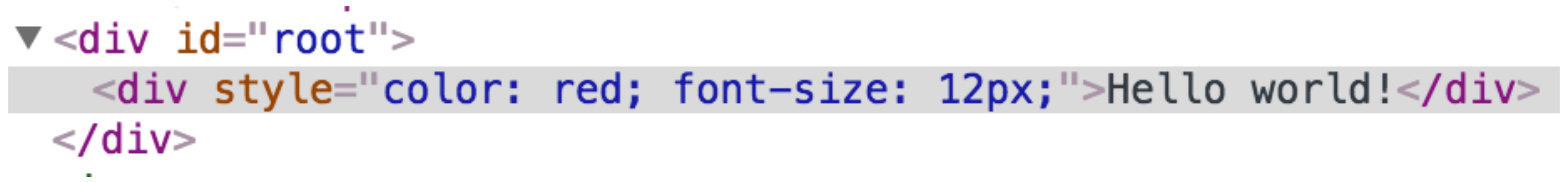
- SASS / LESS / Stylus
- OOCSS / SMACSS / BEM / ITCSS
- CSS Modules
- CSS in JS

React 诞生

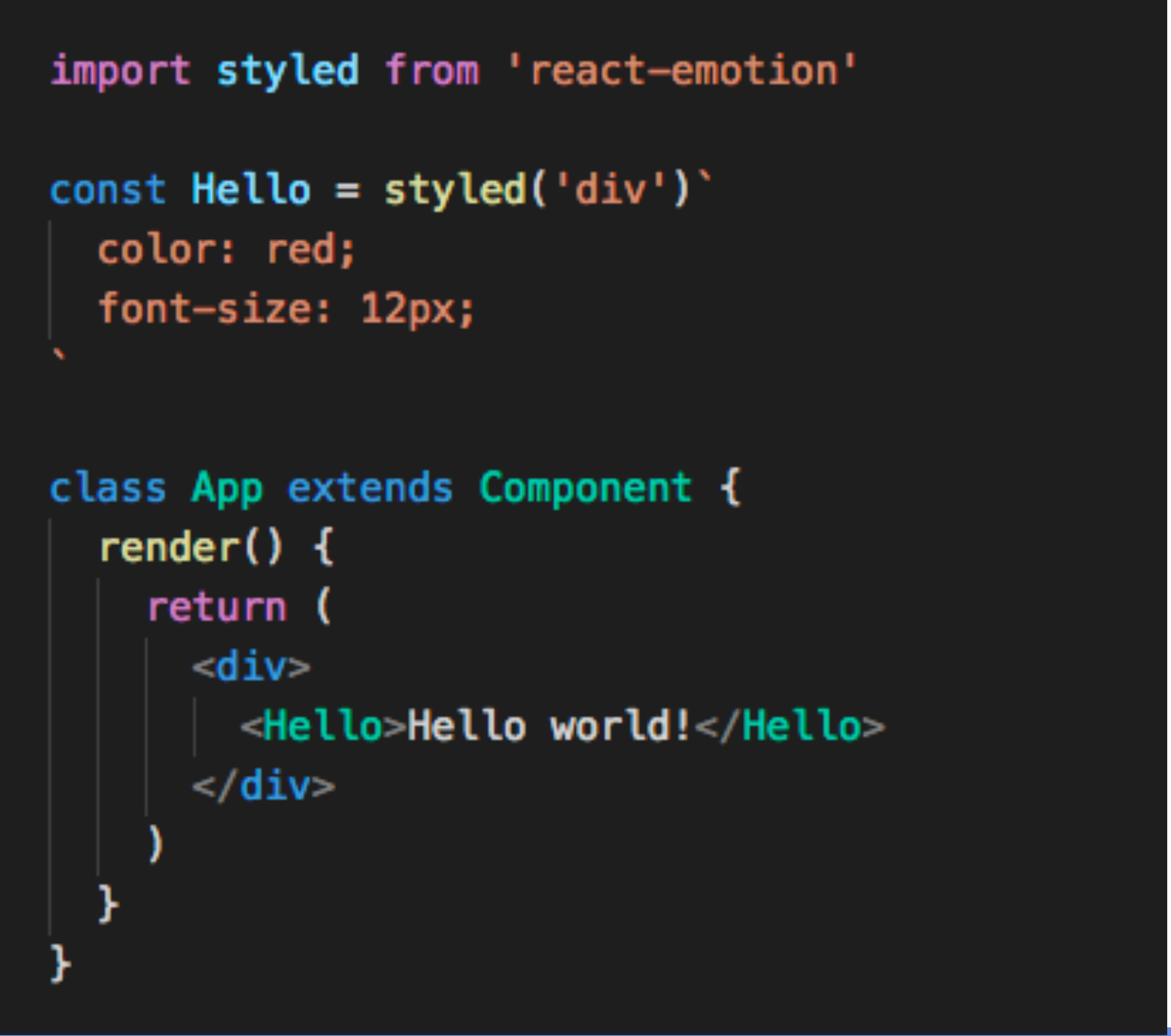


```
const style = {
  'color': 'red',
  'fontSize': '12px'
}

class App extends Component {
  render() {
    return (
      <div style={style}></div>
    );
  }
}
```

Inline style

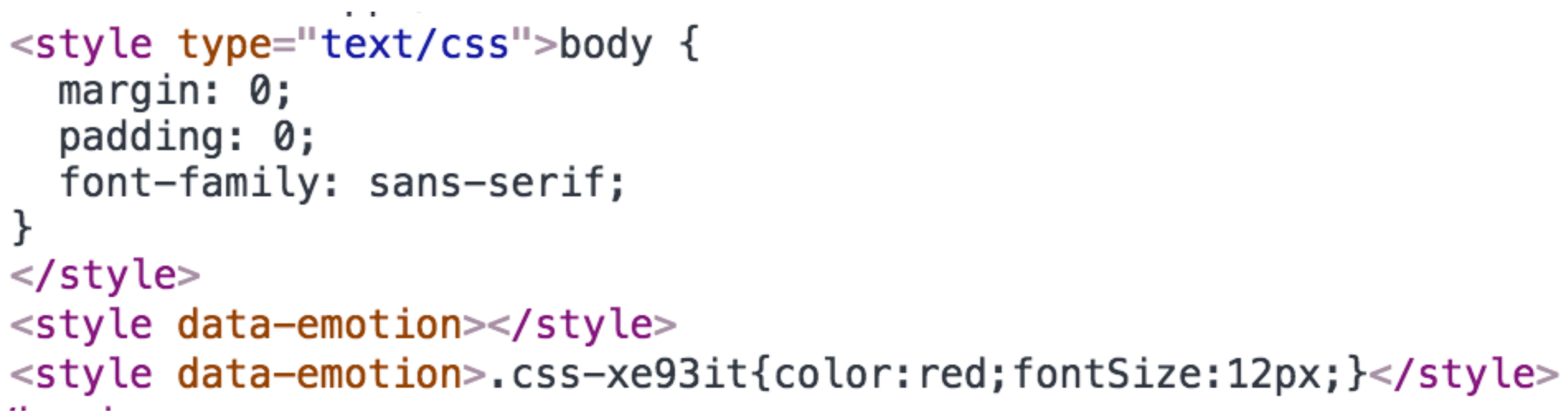


```
▼<div id="root">
   <div style="color: red; font-size: 12px;">Hello world!</div>
 </div>
```

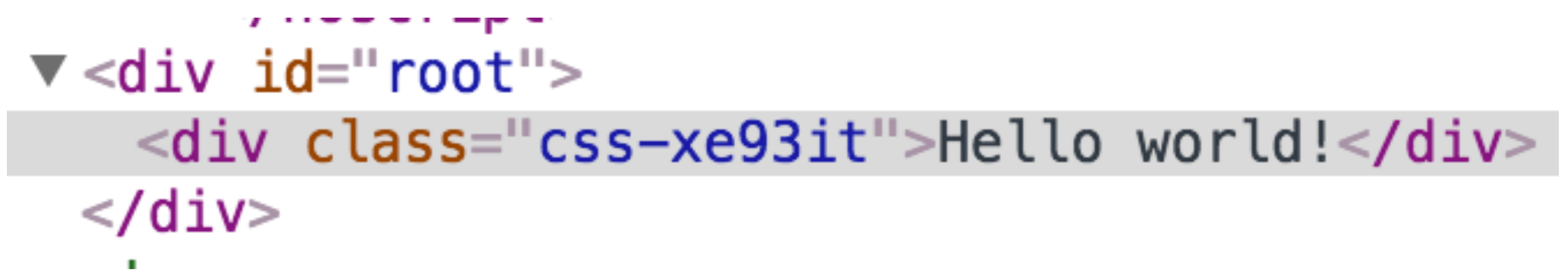


```
import styled from 'react-emotion'

const Hello = styled('div')`
  color: red;
  font-size: 12px;
`


class App extends Component {
  render() {
    return (
      <div>
        <Hello>Hello world!</Hello>
      </div>
    )
  }
}
```

CSS in JS



```
<style type="text/css">body {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
}
</style>
<style data-emotion></style>
<style data-emotion>.css-xe93it{color:red;fontSize:12px;}</style>
```



```
▼ <div id="root">
    <div class="css-xe93it">Hello world!</div>
  </div>
```

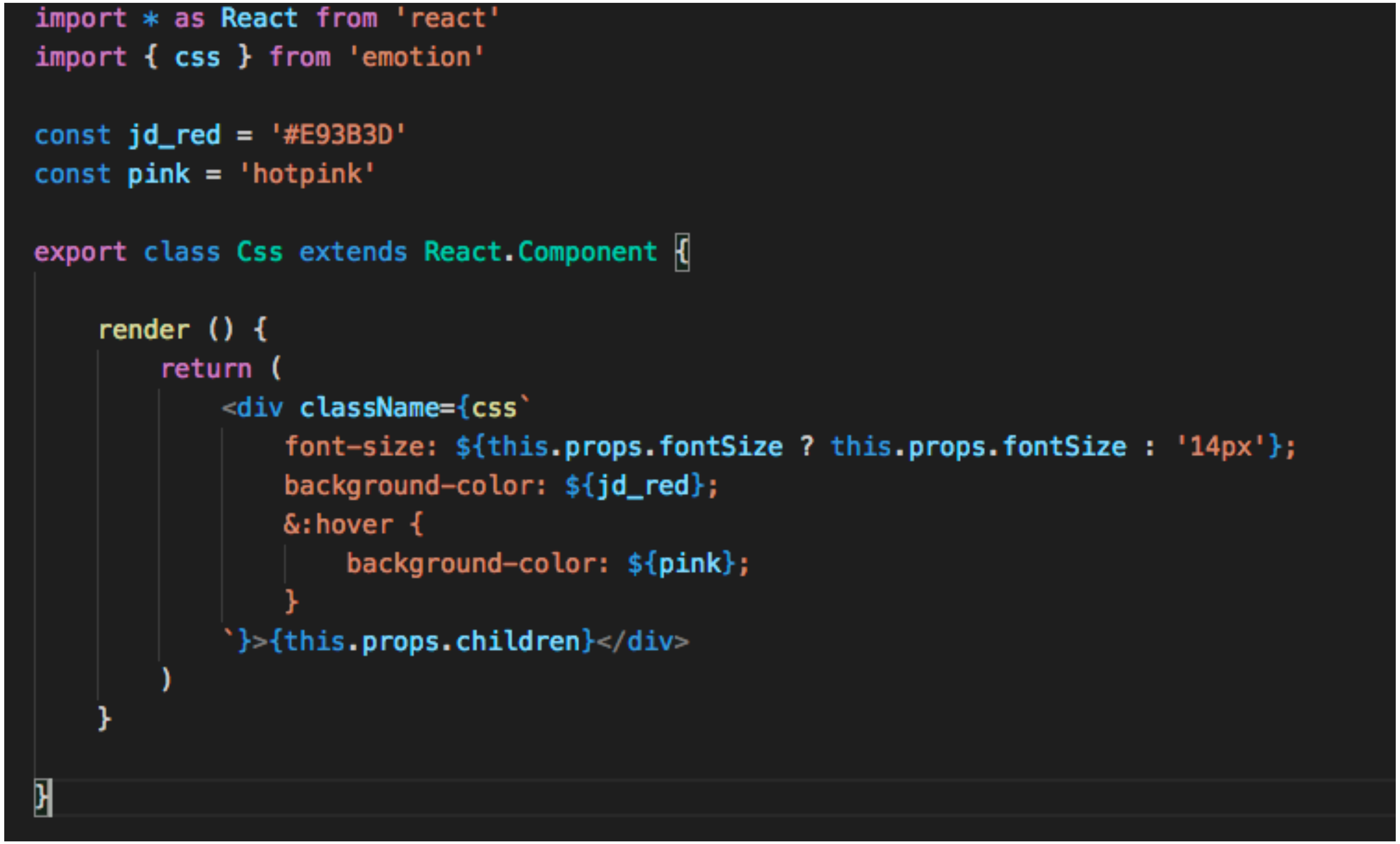
CSS in JS 框架

·styled-components

· glamorous

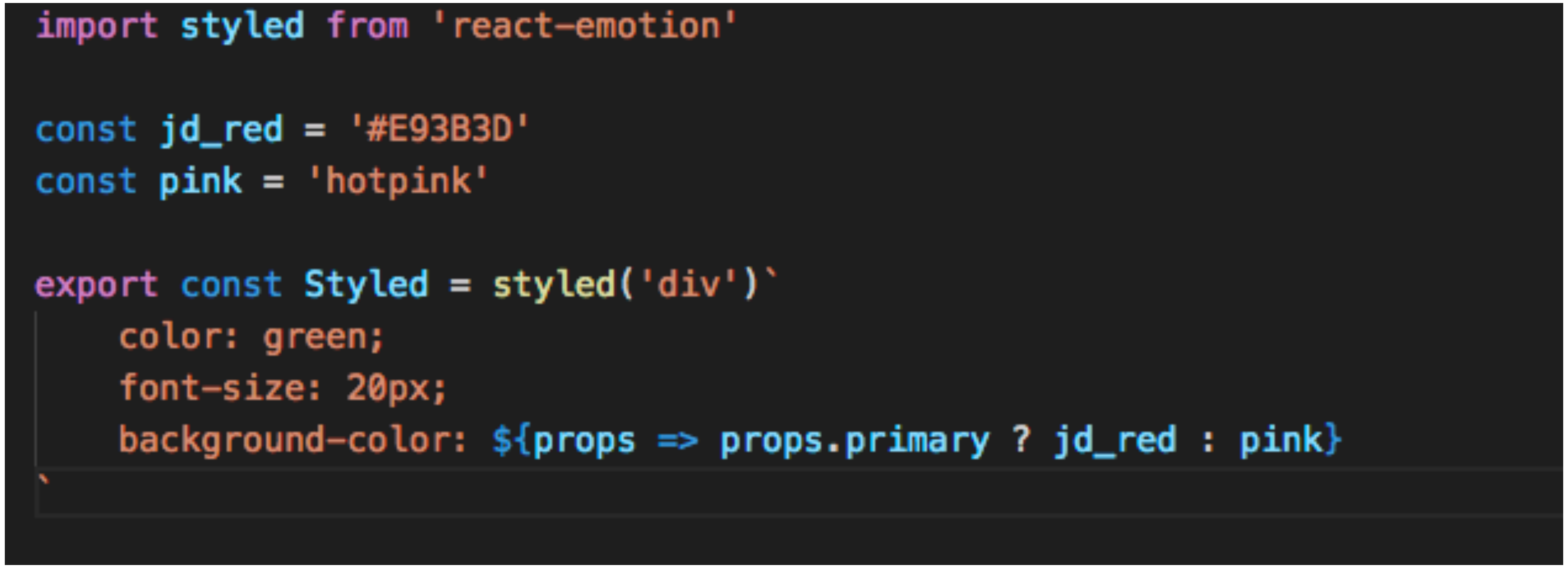
· emotion

emotion



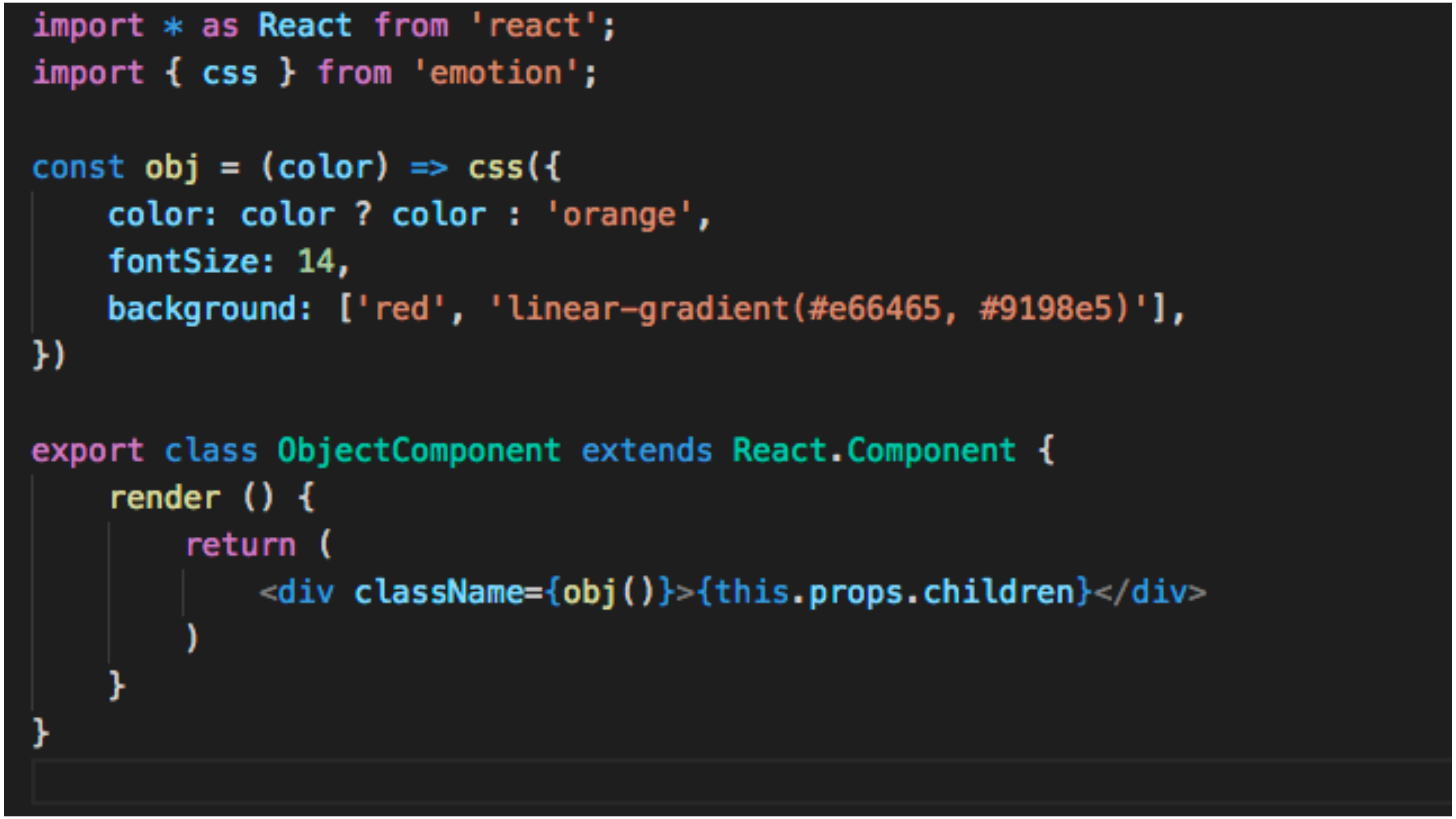
```
import * as React from 'react'
import { css } from 'emotion'

const jd_red = '#E93B3D'
const pink = 'hotpink'

export class Css extends React.Component {

    render () {
        return (
            <div className={css`
                font-size: ${this.props.fontSize ? this.props.fontSize : '14px'};
                background-color: ${jd_red};
                &:hover {
                    background-color: ${pink};
                }
            `}>{this.props.children}</div>
        )
    }

}
```

CSS



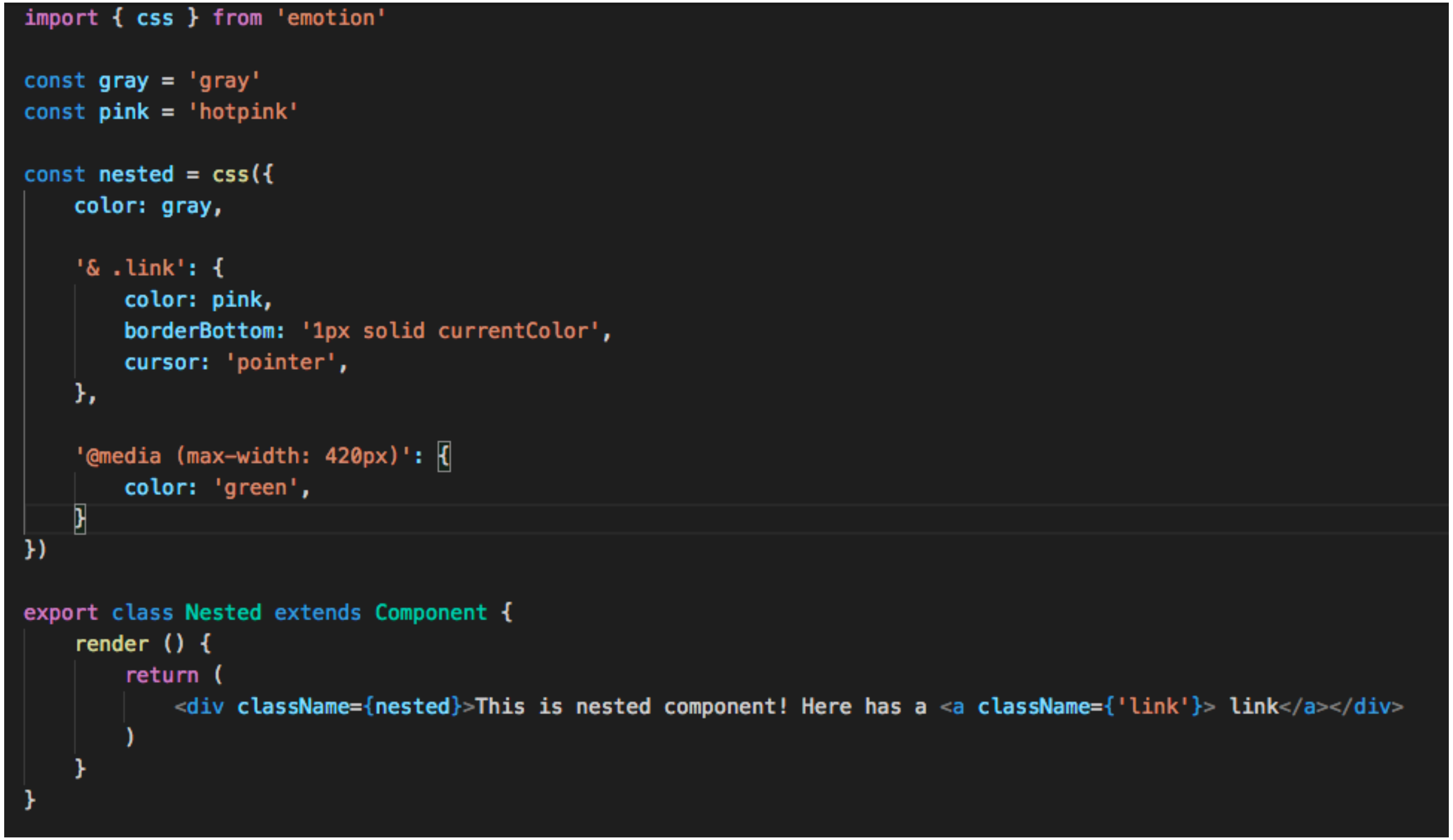
```
import styled from 'react-emotion'

const jd_red = '#E93B3D'
const pink = 'hotpink'

export const Styled = styled('div')`
    color: green;
    font-size: 20px;
    background-color: ${props => props.primary ? jd_red : pink}
`
```

Styled Components



```
import * as React from 'react';
import { css } from 'emotion';

const obj = (color) => css({
    color: color ? color : 'orange',
    fontSize: 14,
    background: ['red', 'linear-gradient(#e66465, #9198e5)'],
})

export class ObjectComponent extends React.Component {
    render () {
        return (
            <div className={obj()}>{this.props.children}</div>
        )
    }
}
```

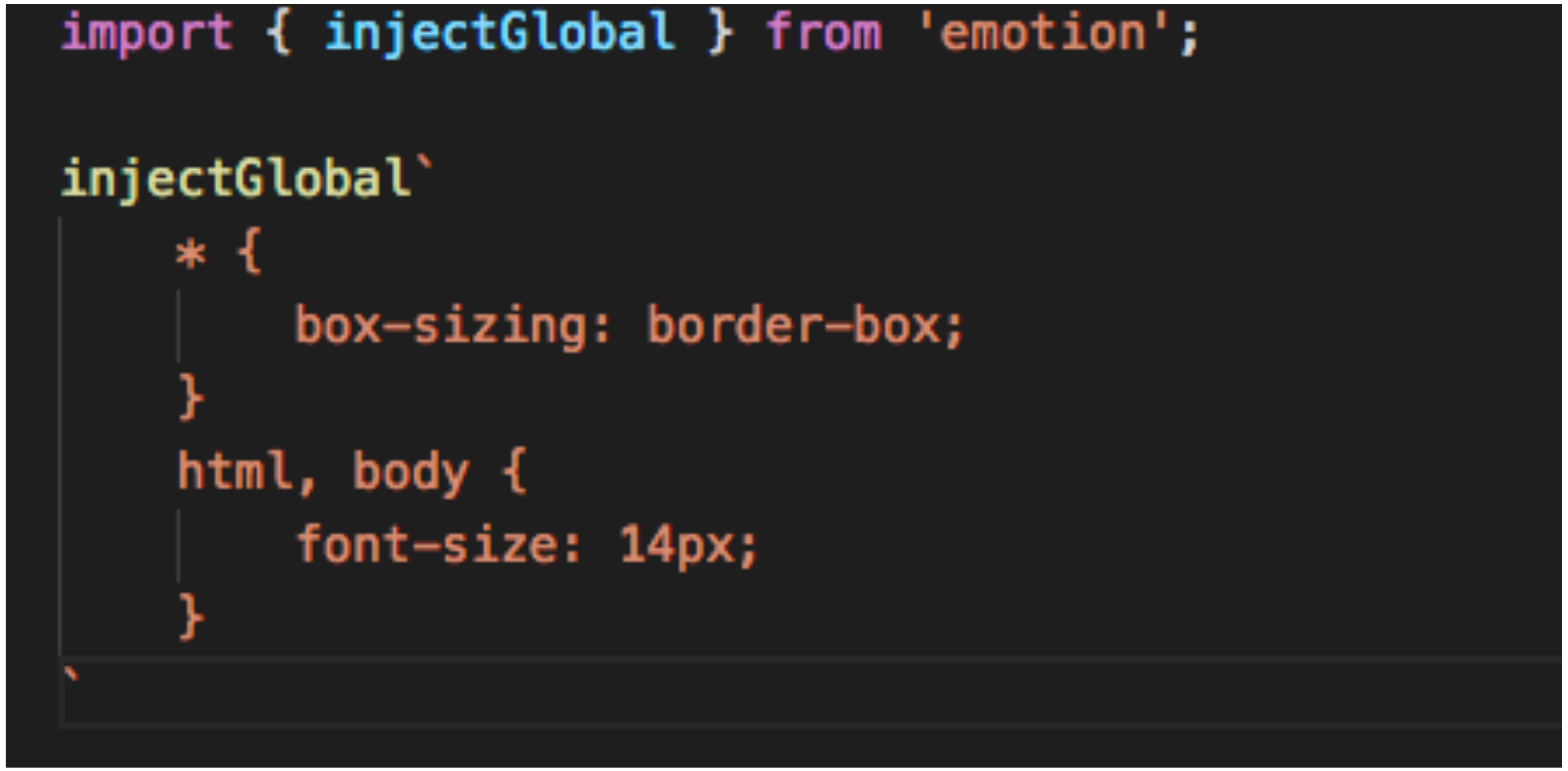
Object Styles



```
import { css } from 'emotion'

const gray = 'gray'
const pink = 'hotpink'

const nested = css({
    color: gray,

    '& .link': {
        color: pink,
        borderBottom: '1px solid currentColor',
        cursor: 'pointer',
    },

    '@media (max-width: 420px)': {
        color: 'green',
    }
})

export class Nested extends Component {
    render () {
        return (
            <div className={nested}>This is nested component! Here has a <a className={'link'}> link</a></div>
        )
    }
}
```

Nested Selectors

Global Styles



```
import { injectGlobal } from 'emotion';

injectGlobal`
    * {
        box-sizing: border-box;
    }
    html, body {
        font-size: 14px;
    }
`
```

优缺点

优点

· 生成唯一 classname，避免全局污染，解决命名规则混乱

· JavaScript 和 CSS 之间变量共享，方便灵活

· 只生成页面所需要的代码，缩减了最终包的大小

· All in JavaScript

· css 单元测试

缺点

· 把 CSS 写进 JS， 增加复杂度，学习成本高

· 对前端框架依赖度高，比如 React

· 覆盖第三方插件样式可能会权重不够

· lint 工具不友好

T H A N K S

FOR YOUR WATCHING

OPEN ORIENTED
凹 凸 实 验 室