

React Moscow  
Meetup #2



Артём Арутюнян @artalar

фанат JS, покоритель DX, любитель OS, CS мечтатель



## Styled Components

библиотека для построения атомарных компонентов отображения

- Знакомство;
- Архитектура;
- Проблемы и решения;
- API;
- DX и окружение;
- Известные проблемы;
- Обзор использования в больших проектах;
- Ссылки на материалы;
- Вопросы.

# Styled Components

styled-components в массах



**16к звёзд на GitHub**



**Более 20 публичных  
production приложений**



**Сотня связанных материалов**

Компоненты, гриды, хелперы, тесты, шаблоны,  
реальные приложения в open source.

# Styled Components

styled-components в массах



**16к звёзд на GitHub**



**Более 20 публичных  
production приложений**



**Сотня связанных материалов**

Компоненты, гриды, хелперы, тесты, шаблоны,  
реальные приложения в open source.

**А вы всё ещё на static build css?**



# История и предпосылки



ES6 и template literals



Компонентный подход



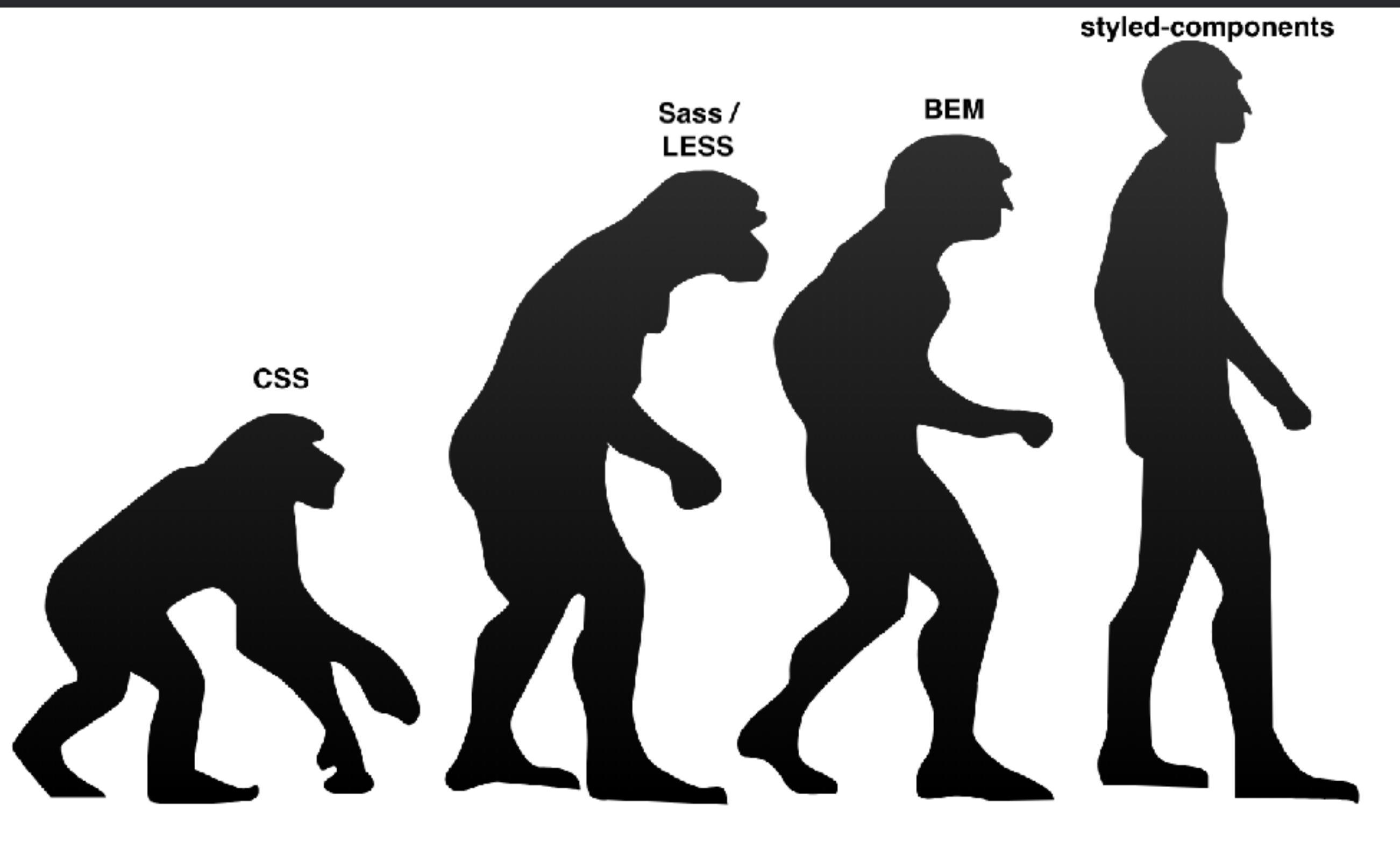
CSS in JS



Страдания от решения  
проблемы динамических  
стилей



*Visual primitives for the component age. Use the best bits of  
ES6 and CSS to style your apps without stress*



# Архитектура

## НОВЫЙ ВЗГЛЯД

- HTML, CSS, JS;
- JSX;
- ООП — это про объекты и их взаимодействие;
- Компонентный подход;
- Абстракция над задачей, а не технологией;
- SC делает атомарные компоненты отображения правильно.



# Архитектура

## НОВЫЙ ВЗГЛЯД

---

```
// Component.js
import cn from "classnames";

export const Component = ({ disabled, children }) => (
  <div
    className={cn(
      "block__element--disabled": disabled,
      block__element: !disabled
    )}
  >
    {children}
  </div>
);

// .sass
.block__element {
  opacity: 1;

  &--disabled {
    opacity: 0.8;
  }
}
```

# Архитектура

## НОВЫЙ ВЗГЛЯД

---

```
// Component.js
import cn from "classnames";

export const Component = ({ disabled, children }) => (
  <div
    className={cn(
      "block__element--disabled": disabled,
      block__element: !disabled
    )}
  >
    {children}
  </div>
);

// .sass
.block__element {
  opacity: 1;

  &--disabled {
    opacity: 0.8;
  }
}
```

```
// Component.js
import styled from "styled-components";

export const Component = styled.div`
  opacity: ${p => p.disabled ? 0.8 : 1};
`;
```



# Архитектура

## НОВЫЙ ВЗГЛЯД

---

```
render() {  
  const { status, onClose } = this.props;  
  const { header, description, tip, button, img } =  
  
  return (  
    <Container>  
      <ContainerBody>  
        <Img src={img} />  
        <Header>{header}</Header>  
        <Description>{description}</Description>  
        <Tip>{tip}</Tip>  
        <Button onClick={onClose}>{button}</Button>  
      </ContainerBody>  
    </Container>  
  );  
}
```

# Новый Взгляд

Styled-components. Архитектура

## Separation of Concerns



# Проблемы и решения

## Проблемы до styled-components

---

1. Динамические стили (через классы или инлайн);
2. Коллизия имен (глобальный скоуп);
3. Сахар: константы, селекторы (Sass, Less нужно учить);
4. Dead CSS (сложно отследить);
5. Critical CSS;
6. Сборка.

# Проблемы и решения

## Решения со styled-components

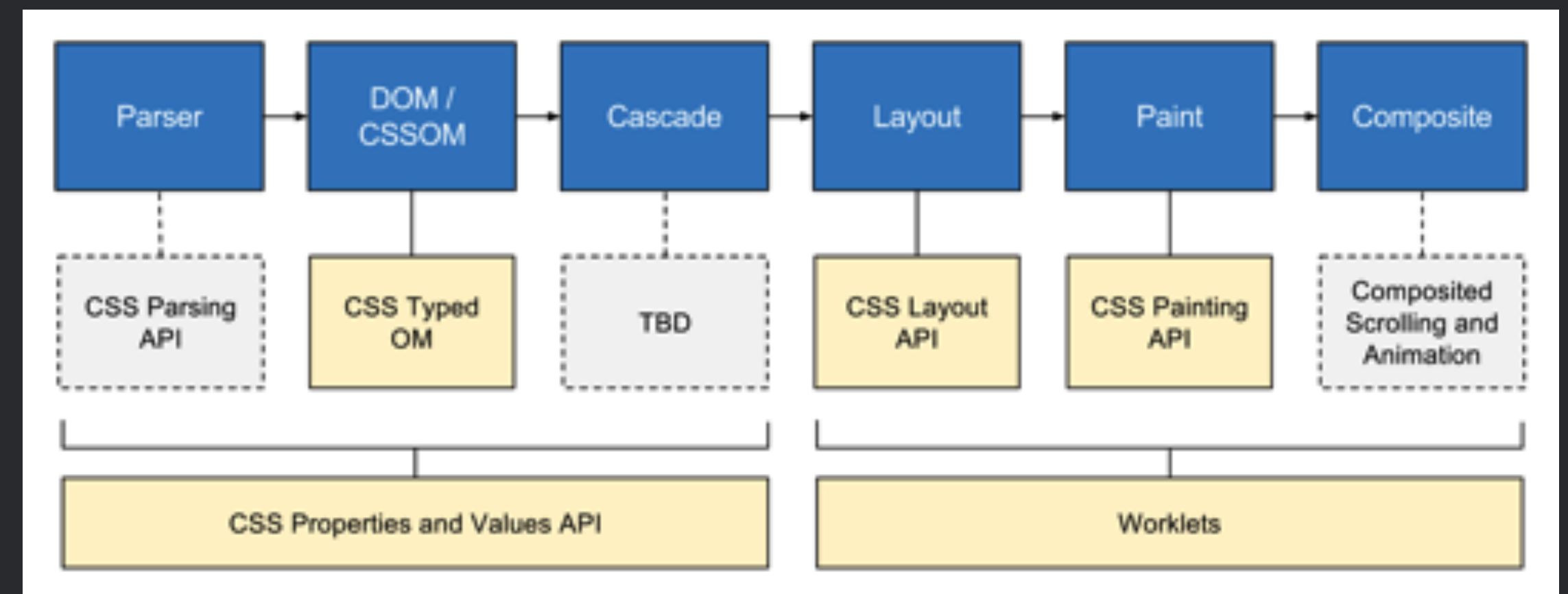
---

1. Стили — функция от аргументов;
2. Динамическая генерация классов (+ их минификация);
3. Вся мощь JS;
4. Компонентный подход;
5. Lazy loading;
6. Runtime.

# Проблемы и решения

## Почему CSS in JS

- Inline-стили (нет);
- Всё, что было сказано выше;
- CSS — это не язык программирования, а язык описания внешнего вида документа;
- Houdini;
- И вообще, это хотели ещё в 2006.



# API

## API “Styled”

```
import styled from 'styled-components';
import ExternalComponent from 'ui-kit';

export const Component = styled.div`
  background-color: palevioletred;
  @media(min-width: 768px) {
    background: mediumseagreen;
  }
`;

export const ExternalWrappedComponent = styled(ExternalComponent)`
  background-color: palevioletred;

  .block__element__modifier {
    color: ${props => props.inner.fontColor};
  }
`;
```



# API

## API “styled.attrs”

```
export const Component = styled.div.attrs({
  style: props => ({ backgroundColor: props.color }),
})`
width: 15rem;
height: 5rem;
`;
```

```
export const Icon = styled.img`
  width: 1rem;
  height: 1rem;
`;

export const Avatar = styled.div`
  padding: ${props => (props.size * 0.2) + 0.8}rem;

  ${Icon} {
    width: ${props => (props.size * 0.2) + 0.8}rem;
    height: ${props => (props.size * 0.2) + 0.8}rem;
  }
`;
```

## Styled Components

API селекторы

- sc-block;
- sc-selector;
- sc-declaration;
- sc-property;
- sc-value.

# API

---

- `styled().extend` deprecated;
- `styled().withComponent`;
- `styled().innerRef`;
- `import { css } from 'styled-components'`;
- `import { keyframes } from 'styled-components'`;
- `import { injectGlobal } from 'styled-components'`;
- `import { withTheme } from 'styled-components'`;
- `import { isStyledComponent } from 'styled-components'`.



# DX и окружение

---

- Документация по экосистеме с “awesome list”;
- Stylelint и подсветка кода;
- babel-plugin-styled-components;
- Yoga, React Native;
- yo-yo, choo.js.

```
▼ <div class="PageWrapper-iDiHMJ hsQA0I">  
  ► <div class="SignIn__Header-cIaTHk dgItRl">...</div>  
  ▼ <form class="FormWrapper-byyWtk lffZeS" action>  
    <div class="SuccessRegistration-erSjmA fCAyyE"></div>
```

# Известные проблемы

---

- Привязка к фреймворку React.js;
- Размер бандла: 42.3kB minified, 15.8 kB minified + gzipped;
- Дебаг миксинов;
- Свойства с зарезервированными названиями.
- Постпроцессоры?
- Типизация?
- Скорость?



# Real world app

## Темизация

- Константы;
- Анимации;
- Миксины;
- Фабрики миксинов;
- polished;
- styled-system;
- Rebass.

```
▼ object {breakpoints: object, direction: "ltr", mixins: object, overrides: object, palette: object..}
  ▼ breakpoints: object
    ▼ keys: Array[5]
      0: "xs"
      1: "sm"
      2: "md"
      3: "lg"
      4: "xl"
    ▼ values: object
      xs: 0
      sm: 600
      md: 960
      lg: 1200
      xl: 1920
      up: function f()
      down: function d()
      between: function p()
      only: function v()
      width: function h()
    direction: "ltr"
  ▼ mixins: object
    gutters: function r()
  ▼ toolbar: object
    minHeight: 50
    ▼ @media (min-width:8px) and (orientation: landscape): object
      minHeight: 48
    ▼ @media (min-width:960px): object
      minHeight: 64
  overrides: object
  ▼ palette: object
    ▼ common: object
      black: "#000"
      white: "#fff"
      type: "dark"
    ▼ primary: object
      light: "#444444"
      main: "#333333"
      dark: "#333333"
      contrastText: "#fff"
    ▼ secondary: object
      light: "#ff4081"
      main: "#ff5057"
      dark: "#c51162"
      contrastText: "#fff"
    ▼ error: object
      light: "#e57373"
      main: "#f44336"
      dark: "#d32f2f"
      contrastText: "#fff"
    ▼ gray: object
      50: "#fafafa"
      100: "#f5f5f5"
      200: "#cccccc"
      300: "#c0c0c0"
      400: "#bdbdbd"
      500: "#9e9e9e"
      600: "#757575"
      700: "#616161"
      800: "#424242"
      900: "#212121"
      A100: "#bdbdbd"
      A200: "#bdbdbd"
      A400: "#303030"
      A700: "#616161"
    contrastThreshold: 3
    getContrastText: function e()
    augmentColor: function f()
    tonalOffset: 0.2
  ▼ text: object
    primary: "#fff"
    secondary: "rgba(255, 255, 255, 0.7)"
```



# Real world app

Темизация

---

- Константы;
- Анимации;
- Миксины;
- Фабрики миксинов;
- polished;
- styled-system;
- Rebass.

```
spacing: {  
  small: css`  
    padding: 0.3rem;  
    margin: 0.3rem;  
  `,  
  normal: css`  
    padding: 0.5rem;  
    margin: 0.5rem;  
  `,  
  big: css`  
    padding: 1rem;  
    margin: 1rem;  
  `,  
},
```

# Real world app

Темизация

---

- Константы;
- Анимации;
- Миксины;
- Фабрики миксинов;
- polished;
- styled-system;
- Rebass.

```
const getSpacing = size => css`  
padding: ${space}rem;  
margin: ${space}rem;  
`;  
  
const [small, normal, big] = [0.3, 0.5, 1].map(getSpacing);  
  
const spacing = { small, normal, big };
```

# Real world app

Темизация

- Константы;
- Анимации;
- Миксины;
- Фабрики миксинов;
- polished;
- styled-system;
- Rebass.

```
// polished
const div = styled.div`
  ${hiDPI(1.5)} {
    width: 200px;
  }
`;

// CSS as JS Output
'@media only screen and (-webkit-min-device-pixel-ratio: 1.5),
only screen and (min--moz-device-pixel-ratio: 1.5),
only screen and (-o-min-device-pixel-ratio: 1.5/1),
only screen and (min-resolution: 144dpi),
only screen and (min-resolution: 1.5dppx)': {
  'width': '200px',
};
```

```
import styled from 'styled-components'
import { width, color } from 'styled-system'

const Box = styled.div`
  ${width}
  ${color}
`;

// width: 50%
<Box width={1/2} />

// font-size: 20px (theme.fontSizes[4])
<Box fontSize={4} />

// color: #333 (theme.colors.gray[0])
<Box color='grays.0' />

// background color
<Box bg='tomato' />
```

# Real world app

Селекторы

---

```
const ifDisabled = props =>
  `
  .props.disabled &&
  .css`
  .opacity: 0.8;
  `;

const backgroundColor = props => css`
  .background-color: ${props.theme.palette[props.color || 'primary']}
  `;

const Component = styled.div`
  .${ifDisabled}
  .${backgroundColor}
  `;
```

# Real world app

Селекторы

---

```
/**
 * @param {(String|Boolean|Function)} cond
 * @param {(Function|Any)} truthy
 * @return {Any}
 */
export const propIf = (condition, truthy) => props =>
  (isfunc(condition) ? condition(props) : isstr(condition) ? !!props[condition] : !!condition)
  ? isfunc(truthy) ? truthy(props) : truthy
  : undefined;
```



# Real world app

Селекторы

---

```
/**
 * Check if cond is true, returns true if exists in props then return truthy of truthy(props)
 * else return falsy or falsy(props)
 *
 * @param {(String|Boolean|Function)} cond
 * @param {(Function|Any)} truthy
 * @param {(Function|Any)} [falsy]
 * @return {Any}
 */
export const propIfElse = (condition, truthy, falsy = undefined) => props =>
  (isfunc(condition) ? condition(props) : isstr(condition) ? !!props[condition] : !!condition)
  ? isfunc(truthy) ? truthy(props) : truthy
  : isfunc(falsy) ? falsy(props) : falsy;
```



# Ссылки на материалы по теме

---

- [Официальная документация](#);
- [Сделано со styled-components](#);
- [Краткая история CSS in JS](#);
- [Про CSS Houdini](#);
- [UI-kit Rebass](#);
- [Вес собранного пакета библиотеки](#);
- [Файл темы в material-ui](#);
- [Описание последнего мажорного обновления styled-components](#).

**Спасибо за внимание**

# Вопросы