

Как покрасить забор если он на React

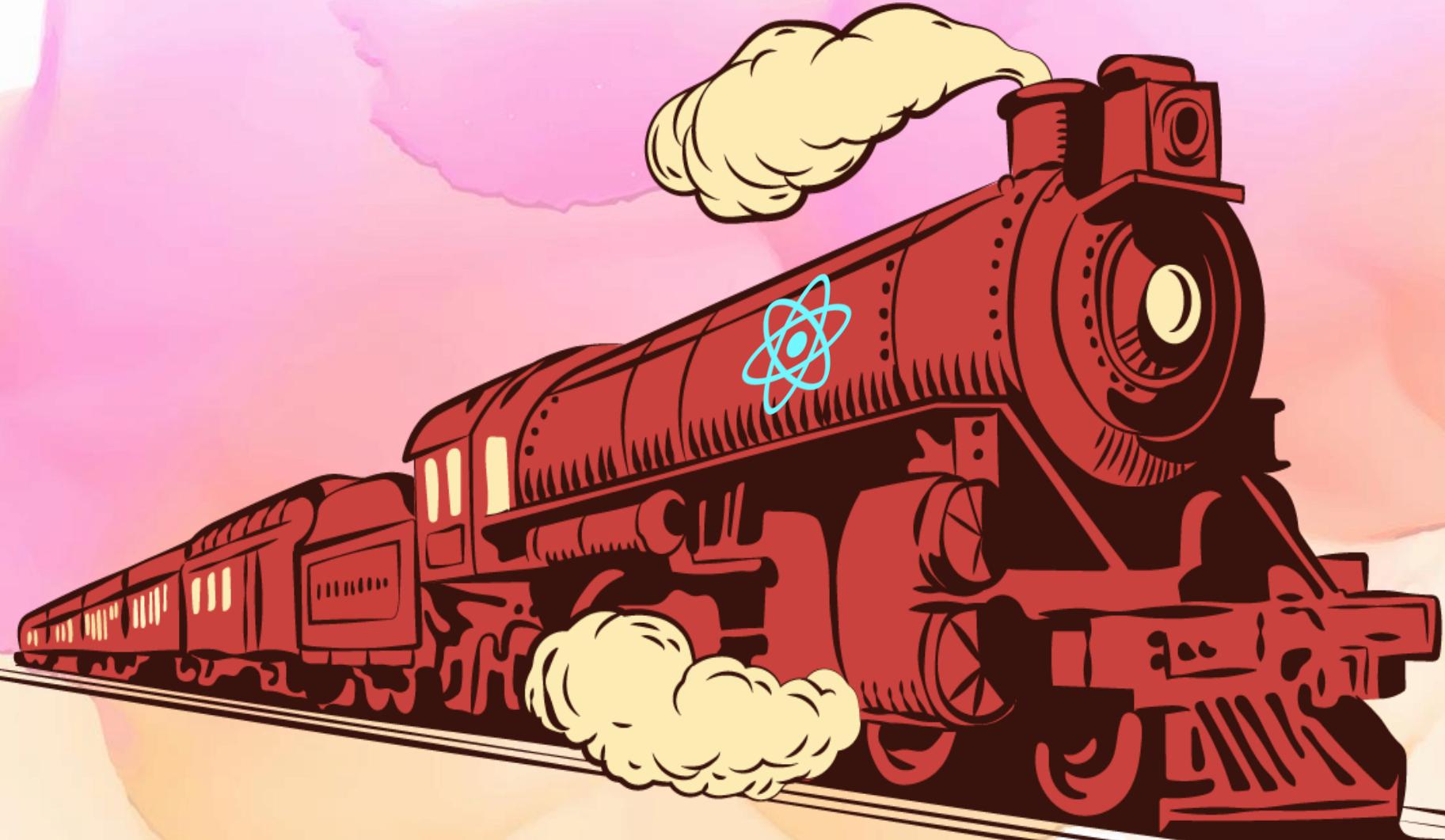


Виктория Тарасевич

Software Engineer – iTechArt Group

[Skype](#) – tarasevich.victoria

[Telegram](#) – @vika_tarasevich

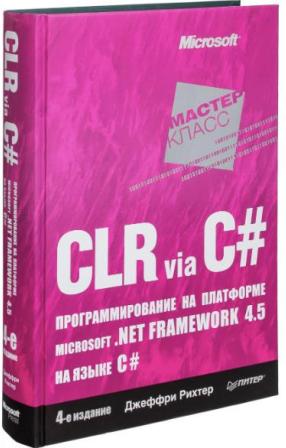




Совершенный код

Стив МакКоннелл

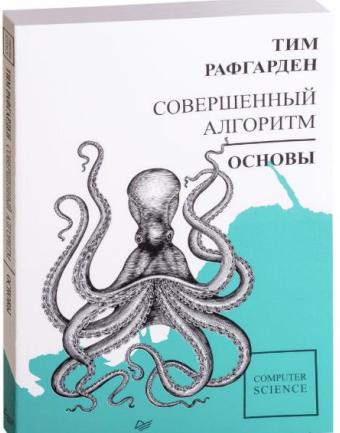
Более 10 лет первое издание этой книги считалось одним из лучших практических руководств по программированию. Сейчас эта книга полностью обновлена с учетом современных тенденций и технологий и дополнена сотнями новых примеров, иллюстрирующих искусство и науку программирования. Опираясь на академические исследования, с одной стороны, и практический опыт коммерческих разработок ПО - с другой, автор синтезировал из самых эффективных методик и наиболее эффективных принципов ясное прагматичное руководство.



CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#

Джеффри Рихтер

Эта книга, выходящая в четвертом издании и уже ставшая классическим учебником по программированию, подробно описывает внутреннее устройство и функционирование общезыковой исполняющей среды (CLR) Microsoft .NET Framework версии 4.5. Написанная признанным экспертом в области программирования Джейфри Рихтером, много лет являющимся консультантом команды разработчиков .NET Framework компании Microsoft, книга научит вас создавать по-настоящему надежные приложения любого вида, в том числе с использованием Microsoft Silverlight, ASP.NET, Windows Presentation Foundation и т. д.



Совершенный алгоритм. Основы

Тим Рафгарден

Алгоритмы - это сердце и душа computer science. Без них не обойтись, они есть везде - от сетевой маршрутизации и расчетов по геномике до криптографии и машинного обучения. «Совершенный алгоритм» превратит вас в настоящего профи, который будет ставить задачи и мастерски их решать как в жизни, так и на собеседовании при приеме на работу в любую IT-компанию.



“Old school”

“Old school”



```
const Book = ({title, description, cover, authors = []}) => (
  <article className="book">
    <img className="cover" src={cover} alt={title}></img>
    <section className="info">
      <h1 className="title">{title}</h1>
      {authors.map(item=><h2 className="author">{item}</h2>)}
      <p className="description">{description}</p>
    </section>
  </article>
)
```

“Old school”

```
● ● ●  
.book {  
    display: flex;  
    height: 300px;  
    border: 1px solid brown;  
    margin: 5px;  
    border-radius: 15px;  
}  
.cover {  
    height: 300px;  
    width: 300px;  
    object-fit: contain;  
}
```

```
● ● ●  
.info {  
    flex-direction: column;  
}  
.title {  
    font-size: 32px;  
    color: maroon;  
}  
.author {  
    font-size: 24px;  
    color: blue;  
}  
.description {  
    color: gray;  
}
```

“Old school”

```
● ● ●  
  
<article class="book">  
    
    <h1 class="title">Совершенный код</h1>  
    <h2 class="author">Стив МакКоннелл</h2>  
    <p class="description">Более 10 лет первое издание этой книги считалось..  
  </p>  
  </section>  
</article>
```

“Old school”

Плюсы

“Old school”

Плюсы

- Легко писать

“Old school”

Минусы

“Old school”

Минусы

Тяжело поддерживать и масштабировать  



“Old school”

Минусы

Тяжело поддерживать и масштабировать  

  Глобальное пространство имен

“Old school”

Минусы

Тяжело поддерживать и масштабировать  

-   Глобальное пространство имен
-  Специфичность

“Old school”

Минусы

Тяжело поддерживать и масштабировать  

-   Глобальное пространство имен
-  Специфичность
-  Каскадность

“Old school”

Минусы

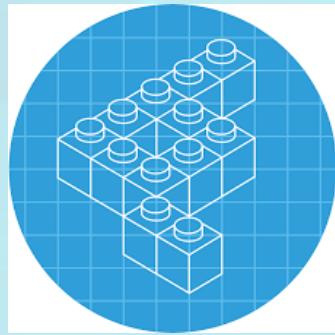
Тяжело поддерживать и масштабировать  

-  Глобальное пространство имен
-  Специфичность
-  Каскадность
-  Мертвый код



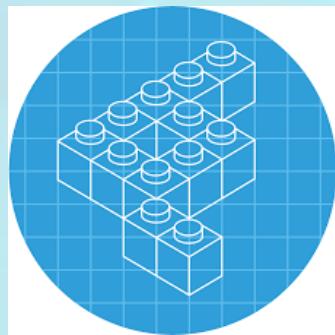
CSS Методологии

CSS Методологии



BEM

CSS Методологии

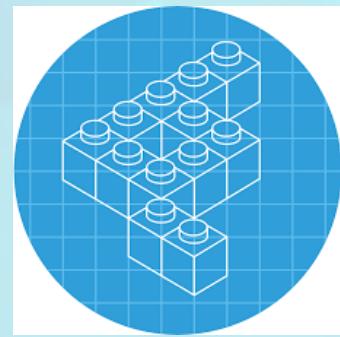


BEM



SMACS
S

CSS Методологии



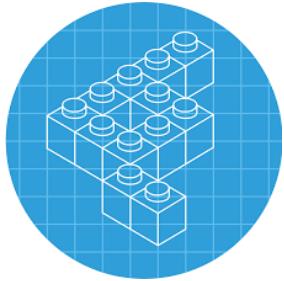
BEM



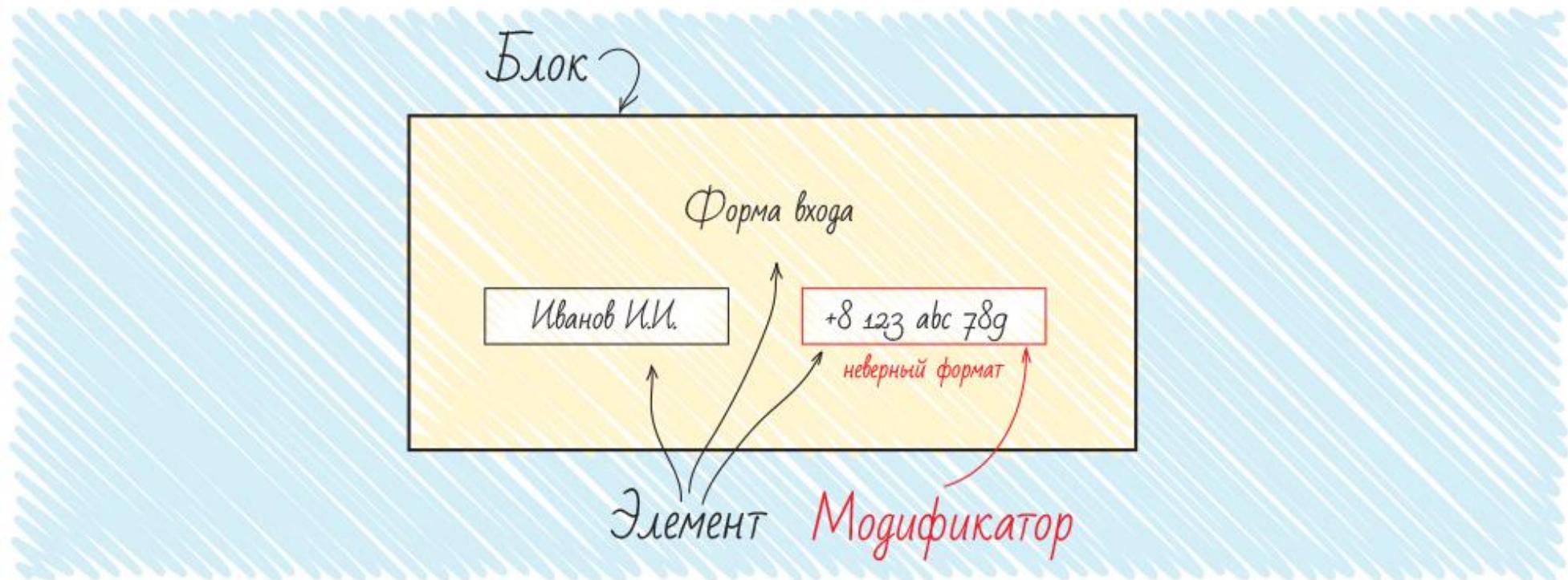
SMACS
S

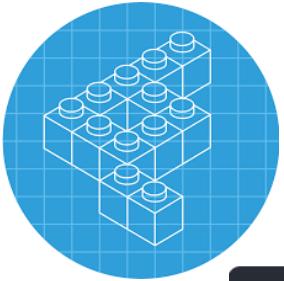


Atomic
CSS



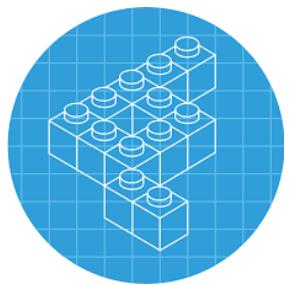
BEM





BEM

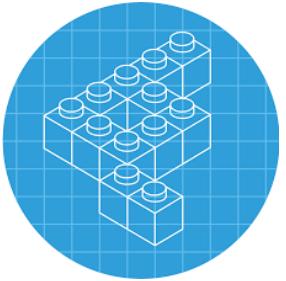
```
const Book = ({title, description, cover, authors = []}) => (
  <article className="book">
    <img className="book__cover" src={cover} alt={title}></img>
    <section className="book__info">
      <h1 className="book__title">{title}</h1>
      {authors.map(item=><h2 className="book__author">{item}</h2>)}
      <p className="book__description">{description}</p>
    </section>
  </article>
)
```



BEM

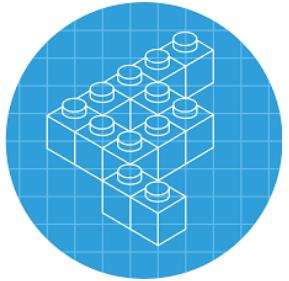
```
● ● ●  
.book {  
    display: flex;  
    height: 300px;  
    border: 1px solid blue;  
    margin: 5px;  
    border-radius: 15px;  
}  
.book__cover {  
    height: 300px;  
    width: 300px;  
    object-fit: contain;  
}
```

```
● ● ●  
.book__info {  
    flex-direction: column;  
}  
.book__title {  
    font-size: 32px;  
    color: green;  
}  
.book__author {  
    font-size: 24px;  
    color: pink;  
}  
.book__description {  
    color: gray;  
}
```



BEM

```
● ● ●  
<article class="book">  
    
  <section class="book__info">  
    <h1 class="book__title">Совершенный код</h1>  
    <h2 class="book__author">Стив МакКоннелл</h2>  
    <p class="book__description">Более 10 лет первое издание этой книги..</p>  
  </section>  
</article>
```



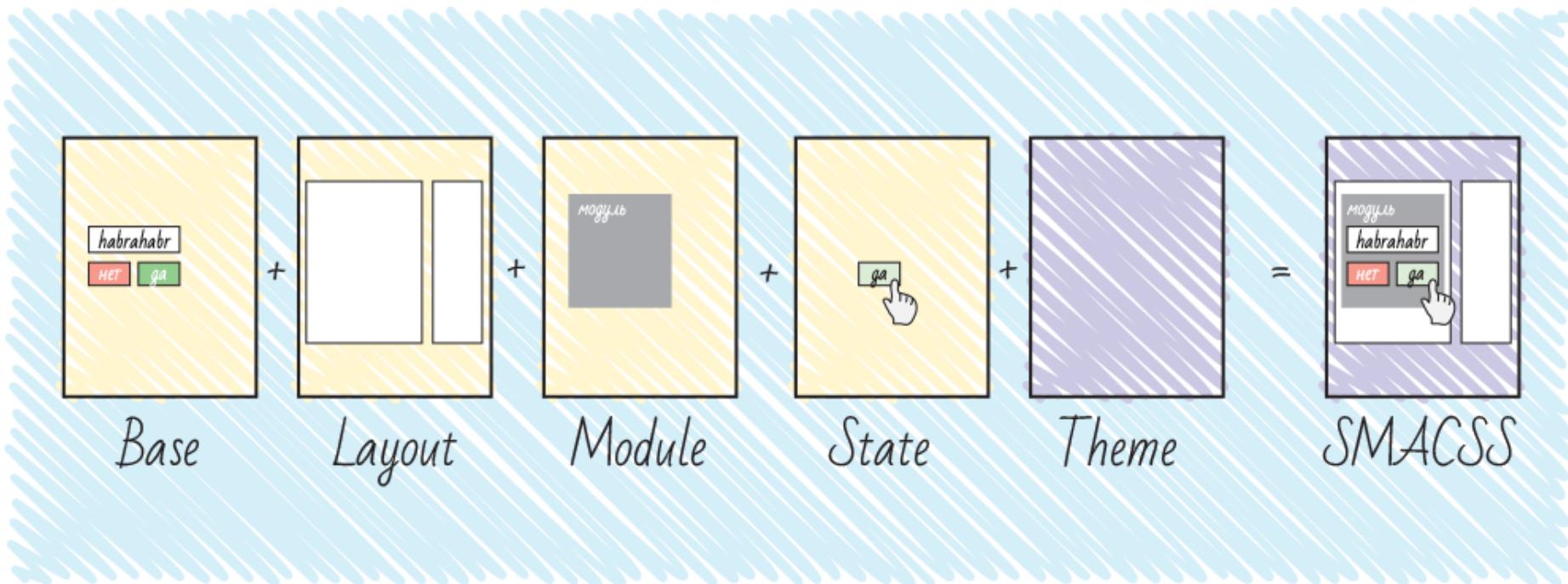
⚠ BEM ⚠



```
<span class="menu__item menu__item_visible menu__item_type_checkbox menu__item_active  
menu__item_size_s">...</span>
```

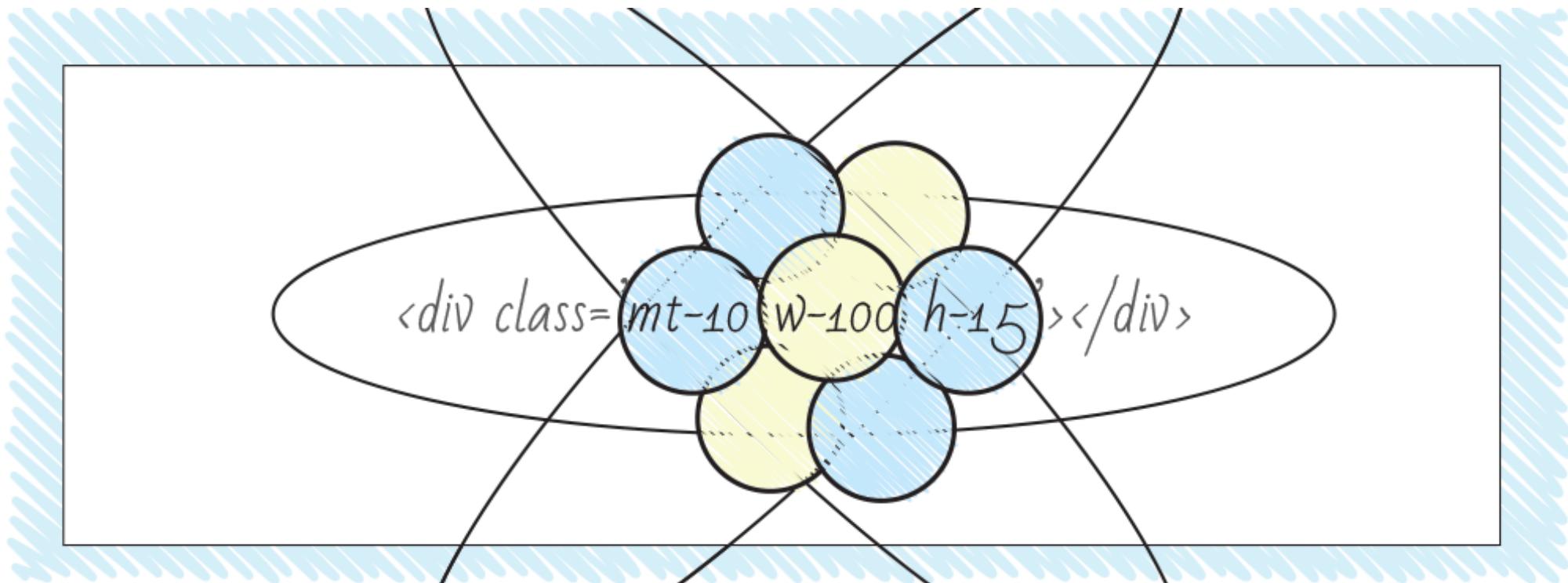


SMACSS





Atomic CSS





Atomic CSS

```
const Book = ({title, description, cover, authors = []}) => (
  <article className="D(f) M(5px) Bdrs(15px) H(300px)">
    <img className="H(300px) W(300px) Objf(ct)" src={cover} alt={title}></img>
    <section className="Fld(c)">
      <h1 className="Fz(32px) C(yellow)">{title}</h1>
      {authors.map(item=><h2 className="Fz(24px) C(gray)">{item}</h2>)}
      <p className="C(pink)">{description}</p>
    </section>
  </article>
)
```



Atomic CSS

```
● ● ●  
.D\f\ {  
    display: flex;  
}  
.M\5px\ {  
    margin: 5px;  
}  
.Fld\c\ {  
    flex-direction: column;  
}  
.Bdrs\15px\ {  
    border-radius: 15px;  
}  
...
```



Atomic CSS

● ● ●

```
<article class="D(f) M(5px) Bdrs(15px) H(300px)">
  
  <section class="Fld(c)">
    <h1 class="Fz(32px) C(yellow)">Совершенный код</h1>
    <h2 class="Fz(24px) C(gray)">Стив МакКоннелл</h2>
    <p class="C(pink)">Более 10 лет первое издание этой книги считалось...</p>
  </section>
</article>
```

CSS Методологии

Плюсы

CSS Методологии

Плюсы

- Читаемость

CSS Методологии

Плюсы

- Читаемость
- Уникальные имена классов

CSS Методологии

Минусы

CSS Методологии

Минусы

 Разметка

CSS Методологии

Минусы

- ➊ Разметка
- ➋ Нет гарантий



CSS

Препроцессоры

CSS Препроцессоры



LESS

CSS Препроцессоры



LESS



SASS

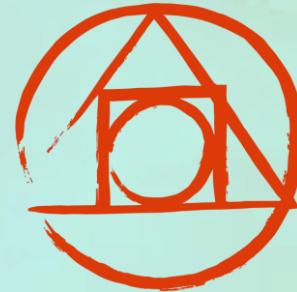
CSS Препроцессоры



LESS



SASS



PostCSS



SASS

```
const Book = ({title, description, cover, authors = []}) => (
  <article className="book">
    <img className="book__cover" src={cover} alt={title}></img>
    <section className="book__info">
      <h1 className="book__title">{title}</h1>
      {authors.map(item=><h2 className="book__author">{item}</h2>)}
      <p className="book__description">{description}</p>
    </section>
  </article>
)
```

Sass

SASS

```
● ● ●  
$primary-color: blue;  
  
.book {  
    display: flex;  
    height: 300px;  
    border: 1px solid $primary-color;  
    margin: 5px;  
    border-radius: 15px;  
  
&__cover {  
    height: 300px;  
    width: 300px;  
    object-fit: contain;  
}  
...  
}
```

Sass

SASS

```
● ● ●  
$primary-color: blue;  
  
.book {  
    display: flex;  
    height: 300px;  
    border: 1px solid $primary-color;  
    margin: 5px;  
    border-radius: 15px;  
  
&__cover {  
    height: 300px;  
    width: 300px;  
    object-fit: contain;  
}  
...  
}
```



SASS

```
● ● ●

<article class="book">
  
  <section class="book__info">
    <h1 class="book__title">Совершенный код</h1>
    <h2 class="book__author">Стив МакКоннелл</h2>
    <p class="book__description">Более 10 лет первое издание этой книги..</p>
  </section>
</article>
```

CSS Препроцессоры

Плюсы

CSS Препроцессоры

Плюсы



Расширение CSS

CSS Препроцессоры

Минусы

CSS Препроцессоры

Минусы

⌚ Время

CSS Модули

CSS Модули

```
import styles from './styles.module.scss'

const Book = ({title, description, cover, authors = []}) => (
  <article className={styles.book}>
    <img className={styles.book__cover} src={cover} alt={title}></img>
    <section className={styles.book__info}>
      <h1 className={styles.book__title}>{title}</h1>
      {authors.map(item=><h2 className={styles.book__author}>{item}</h2>)}
      <p className={styles.book__description}>{description}</p>
    </section>
  </article>
)
```

CSS Модули

```
$primary-color: blue;

.book {
    display: flex;
    height: 300px;
    border: 1px solid $primary-color;
    margin: 5px;
    border-radius: 15px;

    &__cover {
        height: 300px;
        width: 300px;
        object-fit: contain;
    }
    ...
}
```

CSS Модули

```
$primary-color: blue;

.book {
    display: flex;
    height: 300px;
    border: 1px solid $primary-color;
    margin: 5px;
    border-radius: 15px;

    &__cover {
        height: 300px;
        width: 300px;
        object-fit: contain;
    }
    ...
}
```

CSS Модули

```
● ● ●  
  
<article class="styles_book__1fdcM">  
    
    <h1 class="styles_book__title__22aes">Совершенный код</h1>  
    <h2 class="styles_book__author__2RN4r">Стив МакКоннелл</h2>  
    <p class="styles_book__description__19xQ4">Более 10 лет первое издание этой  
      книги..</p>  
  </section>  
</article>
```

CSS Модули

```
● ● ●  
{  
  "book": "styles_book__1fdcM",  
  "book__cover": "styles_book__cover__A2sF-",  
  "book__info": "styles_book__info__3hKEJ",  
  "book__title": "styles_book__title__22aes",  
  "book__author": "styles_book__author__2RN4r",  
  "book__description": "styles_book__description__19xQ4"  
}
```

CSS Модули

Плюсы

CSS Модули

Плюсы



Локальная область видимости

CSS Модули

Минусы

CSS Модули

Минусы

Имена классов



Немного динамики

Немного динамики

```
import clsx from 'clsx';

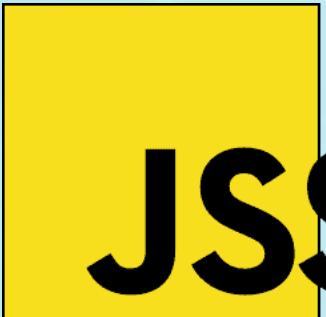
import styles from './styles.module.scss'

const Book = ({title, description, cover, publicationYear, authors = []}) => {
    let bookClasses = clsx(styles.book, { [styles.book_fresh]: publicationYear === new Date().getFullYear() });

    return <article className={bookClasses}>
        <img className={styles.book__cover} src={cover} alt={title}></img>
        <section className={styles.book__info}>
            <h1 className={styles.book__title}>{title}</h1>
            {authors.map(item=><h2 className={styles.book__author}>{item}</h2>)}
            <p className={styles.book__description}>{description}</p>
        </section>
    </article>
}
```

css in JS

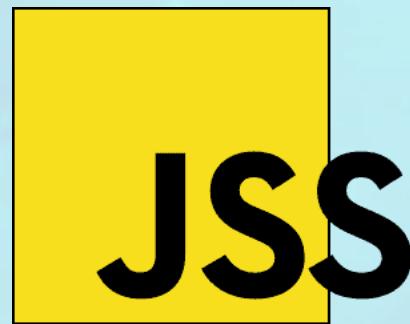
CSS in JS



JSS

JSS

CSS in JS

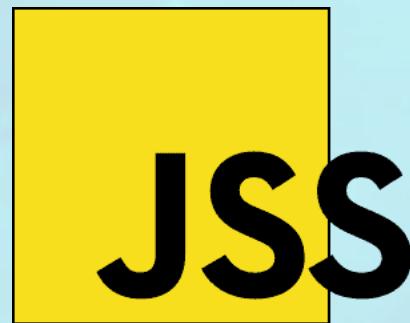


JSS



Emotio
n

CSS in JS



JSS



Emotio
n



Linaria

```
● ● ●

import useStyles from './styles'

const Book = ({title, description, cover, authors = []}) => {
  const classes = useStyles()
  return <article className={classes.book}>
    <img className={classes.cover} src={cover} alt={title}></img>
    <section className={classes.info}>
      <h1 className={classes.title}>{title}</h1>
      {authors.map(item=><h2 className={classes.author}>{item}</h2>)}
      <p className={classes.description}>{description}</p>
    </section>
  </article>
}
```

```
● ● ●

import useStyles from './styles'

const Book = ({title, description, cover, authors = []}) => {
  const classes = useStyles()
  return <article className={classes.book}>
    <img className={classes.cover} src={cover} alt={title}></img>
    <section className={classes.info}>
      <h1 className={classes.title}>{title}</h1>
      {authors.map(item=><h2 className={classes.author}>{item}</h2>)}
      <p className={classes.description}>{description}</p>
    </section>
  </article>
}
```

JSS

JSS

```
● ● ●  
import { createUseStyles } from 'react-jss'  
  
const useStyles = createUseStyles({  
  book: {  
    display: 'flex',  
    height: 300,  
    border: `1px solid ${primaryColor}`,  
    margin: 5,  
    borderRadius: 15,  
  },  
  ...  
})
```

JSS

JSS

```
● ● ●

import { createUseStyles } from 'react-jss'

const useStyles = createUseStyles({
  book: {
    display: 'flex',
    height: 300,
    border: `1px solid ${primaryColor}`,
    margin: 5,
    borderRadius: 15,
  },
  ...
})
```



```
<article class="book-0-2-1">
  
  <section class="info-0-2-3">
    <h1 class="title-0-2-4">Совершенный код</h1>
    <h2 class="author-0-2-5">Стив МакКоннелл</h2>
    <p class="description-0-2-6">Более 10 лет первое издание этой книги...</p>
  </section>
</article>
```



```
<article class="book-0-2-1">
  
  <section class="info-0-2-3">
    <h1 class="title-0-2-4">Совершенный код</h1>
    <h2 class="author-0-2-5">Стив МакКоннелл</h2>
    <p class="description-0-2-6">Более 10 лет первое издание этой книги...</p>
  </section>
</article>
```

```
{  
  "book": "book-0-2-1",  
  "cover": "cover-0-2-2",  
  "info": "info-0-2-3",  
  "title": "title-0-2-4",  
  "author": "author-0-2-5",  
  "description": "description-0-2-6"  
}
```



Немного динамики

```
import useStyles from './styles'

const Book = ({title, description, cover, authors = [], ...otherProps}) => {
  const classes = useStyles(otherProps)
  return <article className={classes.book}>
    <img className={classes.cover} src={cover} alt={title}></img>
    <section className={classes.info}>
      <h1 className={classes.title}>{title}</h1>
      {authors.map(item=><h2 className={classes.author}>{item}</h2>)}
      <p className={classes.description}>{description}</p>
    </section>
  </article>
}
```

```
● ● ●

const useStyles = createUseStyles({
  book: {
    display: 'flex',
    height: 300,
    border: props => `1px solid
      ${props.publicationYear === new Date().getFullYear()
        ? primaryColor
        : secondaryColor
      }`,
    margin: 5,
    borderRadius: 15,
  },
  ...
})
```



Emotion

```
● ● ●  
const Book = ({title, description, cover, publicationYear, authors = []}) => {  
  return <Panel>  
    <Cover src={cover} alt={title}/>  
    <Section>  
      <Title publicationYear={publicationYear}>{title}</Title>  
      {authors.map(item=><Author key={item}>{item}</Author>)}  
      <Description>{description}</Description>  
    </Section>  
  </Panel>  
}
```



Emotion

```
● ● ●

const Book = ({title, description, cover, publicationYear, authors = []}) => {
  return <Panel>
    <Cover src={cover} alt={title}/>
    <Section>
      <Title publicationYear={publicationYear}>{title}</Title>
      {authors.map(item=><Author key={item}>{item}</Author>)}
      <Description>{description}</Description>
    </Section>
  </Panel>
}
```



Emotion



```
import styled from '@emotion/styled'

const Title = styled.h1(props => ({
  fontSize: 32,
  color: 'green',
  '&:hover': {
    color : `${props.publicationYear === new Date().getFullYear() ? primaryColor : secondaryColor}`
  }
}))
```



Emotion



```
import styled from '@emotion/styled'

const Title = styled.h1(props => ({
  fontSize: 32,
  color: 'green',
  '&:hover': {
    color : `${props.publicationYear === new Date().getFullYear() ? primaryColor : secondaryColor}`
  }
}))
```



Emotion

● ● ●

```
<article class="css-letgdv-Panel egt6vrw5">
  
  <section class="css-jwlj7u-Section egt6vrw1">
    <h1 class="css-1fbpsit-Title egt6vrw0">Совершенный код</h1>
    <h2 class="css-slo9jo-Author egt6vrw4">Стив МакКоннелл</h2>
    <p class="css-zvkq9u-Description egt6vrw3">Более 10 лет первое издание этой книги
считалось...</p>
  </section>
</article>
```



А что внутри?

CSS in JS



```
let styleEl = document.createElement('style')
document.head.appendChild(styleEl)

let sheet = styleEl.sheet
sheet.insertRule(`.book {
    display: flex;
    height: 300px;
    border: 1px solid ${primaryColor};
    margin: 5px;
    border-radius: 15px; }`, 0)
```

CSS in JS

```
● ● ●  
  
let styleEl = document.createElement('style')  
document.head.appendChild(styleEl)  
  
let sheet = styleEl.sheet  
sheet.insertRule(`.book {  
    display: flex;  
    height: 300px;  
    border: 1px solid ${primaryColor};  
    margin: 5px;  
    border-radius: 15px; }`, 0)
```



Linaria

```
import { styled } from 'linaria/react';
```

```
const primaryColor = 'burlywood'
```

```
const Title = styled.h1`  
  color: ${primaryColor};  
`;
```



Linaria

```
● ● ●  
  
const Title = styled('h1')({  
  name: 'Title',  
  class: 'Title_tlugh8t',  
  vars: {  
    'tlugh8t9-0': [primaryColor],  
  },  
});
```



Linaria

```
● ● ●  
.Title_tlugh8t9 {  
  font-family: var(--tlugh8t-0);  
}
```

CSS in JS

Плюсы

CSS in JS

Плюсы

- Это реальный CSS

CSS in JS

Плюсы

- Это реальный CSS
- Уникальные имена классов

CSS in JS

Плюсы

- Это реальный CSS
- Уникальные имена классов
- Можно переиспользовать

CSS in JS

Плюсы

- Это реальный CSS
- Уникальные имена классов
- Можно переиспользовать
- Динамические стили

CSS in JS

Минусы

CSS in JS

Минусы

❗ Производительность

CSS in JS

Минусы

- ❗ Производительность
- ❗ Инструменты

CSS in JS

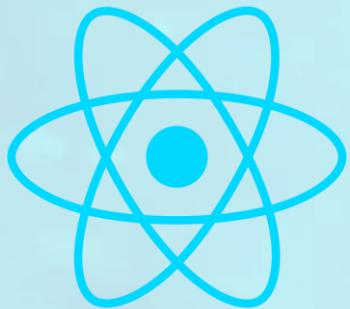
Минусы

- ❗ Производительность
- ❗ Инструменты
- ❗ Безопасность



Так что же
выбрать?

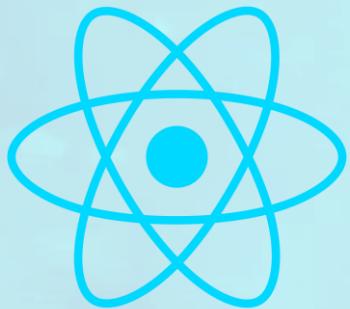
CSS in JS



React Native

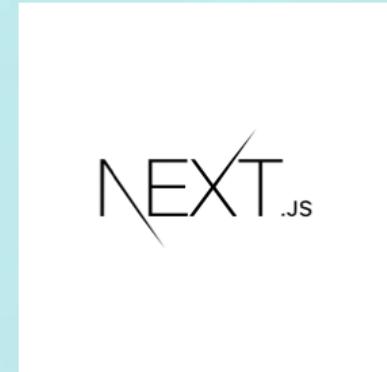
Mobile

CSS in JS



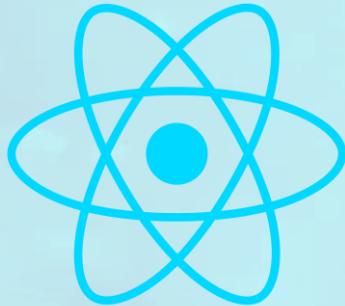
React Native

Mobile



SSR

CSS in JS



React Native

Mobile

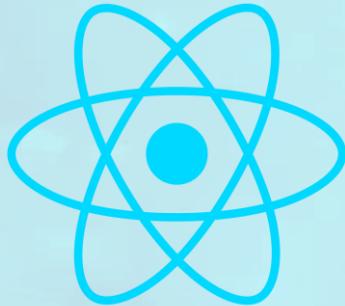


SSR



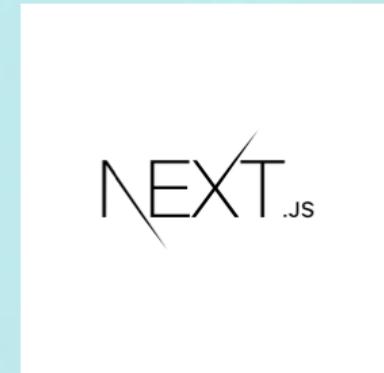
Componen
t library

CSS in JS



React Native

Mobile



SSR



Componen
t library

Спасибо