Stylings

FPT Software Academy

Fpt Software

Contents

1. CSS Overview
2. CSS in JS
3. CSS Module

1. CSS Overview

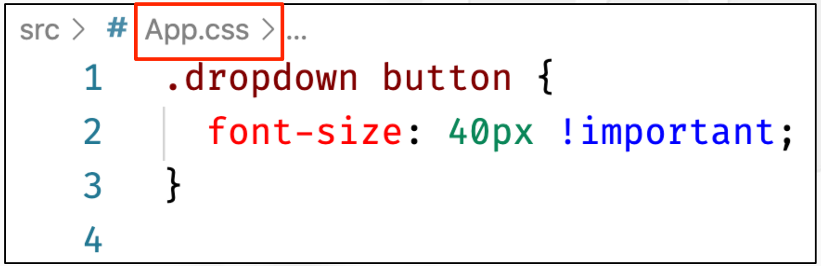
Fpt Software

1. CSS Overview

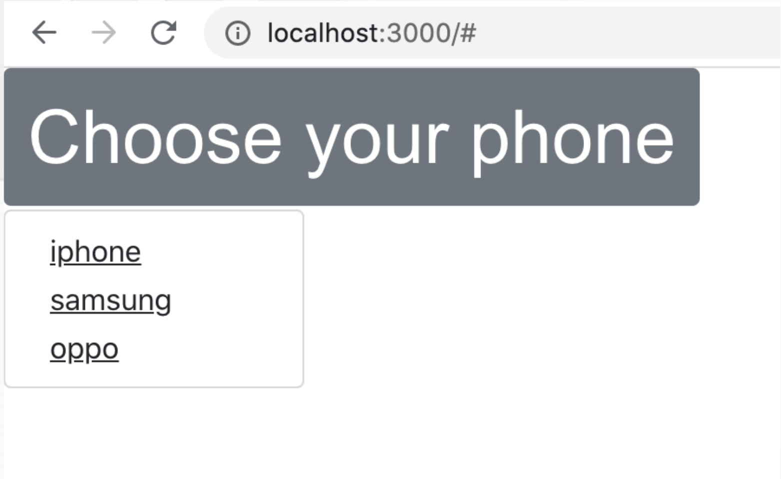
- Recall what we have done with Dropdown Component?
  - Wrap HTML and JS code insode Dropdown.tsx
  - Stylings are in Dropdown.css file
- Do you notice the problem ?
  - CSS for Dropdown Component are globals

1. CSS Overview

CSS in App can easily overwrite CSS of Dropdown



```css
.dropdown button {
  font-size: 40px !important;
}
```



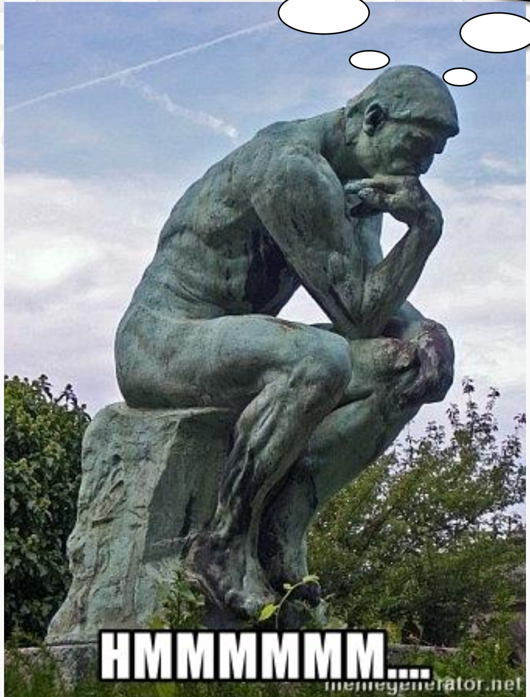
1. CSS Overview

- Our Definition of Component are flawed
  - Component must encapsulate its HTML, CSS, JS code
  - Every code changes are locals
  - HTML,CSS,JS code are not effected by others

1. CSS Overview

How can i prevent other from change CSS of my Component ?

How to make CSS local to Component ?



1. CSS Overview

- Luckily, there are already solutions for this problem:
  - Inline CSS
  - CSS in JS (Styled Component)
  - CSS Module

1. Summary

- CSS styles are global by defauts
- CSS rule can be easily replaced by other rule if not doing correctly
- React does not provides a way to make local CSS, we have to use 3rd party library
- There are 2 most popular approach for local CSS in React:
  - CSS in JS (Styled Component)
  - CSS Module

2. CSS in JS

2. CSS in JS

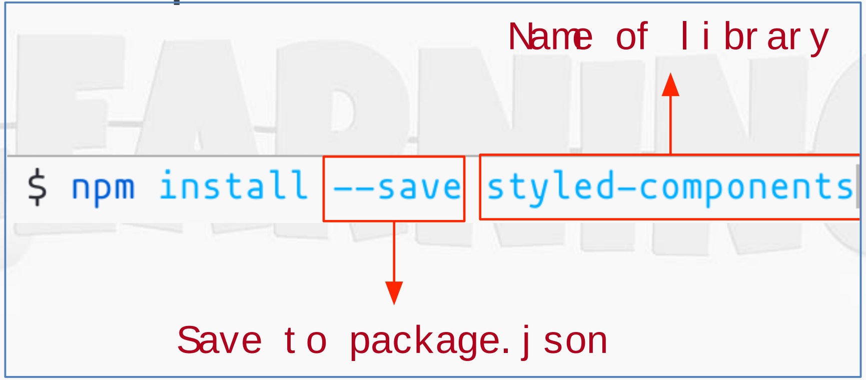
- What is CSS in JS ? Should CSS in its external file ?
  - Refers to a pattern where CSS is composed using JavaScript instead of defined in external files.
  - JavaScript library that bundles each JavaScript component with all its belonging CSS rules and dependencies (Styled Component)
  - As a result, components can run independently, without relying on any external CSS files

2. CSS in JS

- What are benefits of Styled Components ?
  1. Automatic critical CSS
  2. No class name bugs
  3. Easier deletion of CSS
  4. Simple dynamic styling
  5. Painless maintenance
  6. Automatic vendor prefixing

2. CSS in JS

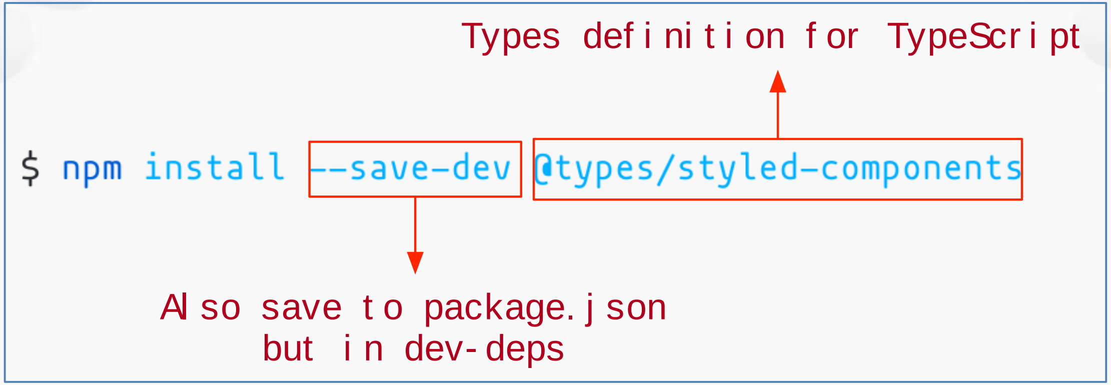
- Setup Styled Component in React:



```
$ npm install --save styled-components
```

Name of library

Save to package.json



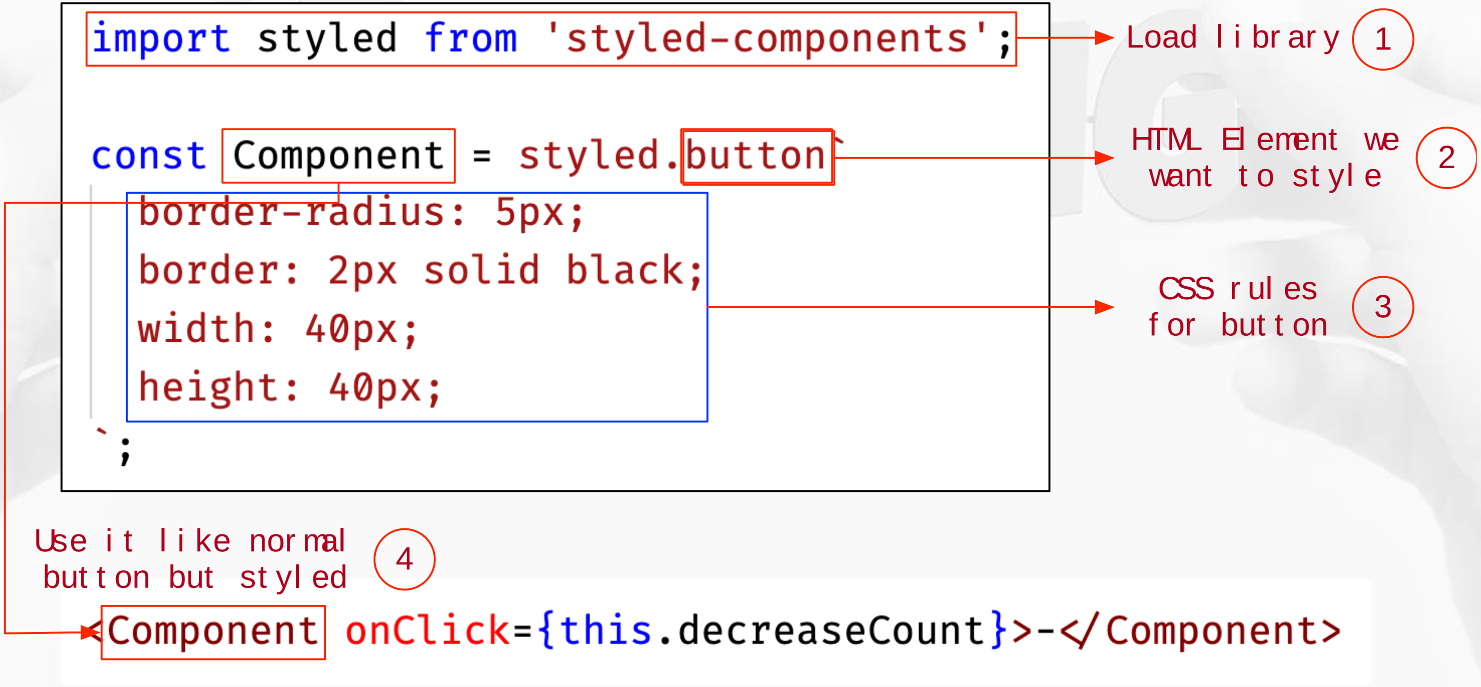
```
$ npm install --save-dev @types/styled-components
```

Types definition for TypeScript

Also save to package.json
but in dev-deps

2. CSS in JS

- Syntax of Styled Component:



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

const Component = styled.button`
  border-radius: 5px;
  border: 2px solid black;
  width: 40px;
  height: 40px;
`;
```

Load library 1

HTML Element we want to style 2

CSS rules for button 3

Use it like normal button but styled 4

```
<Component onClick={this.decreaseCount}>-</Component>
```

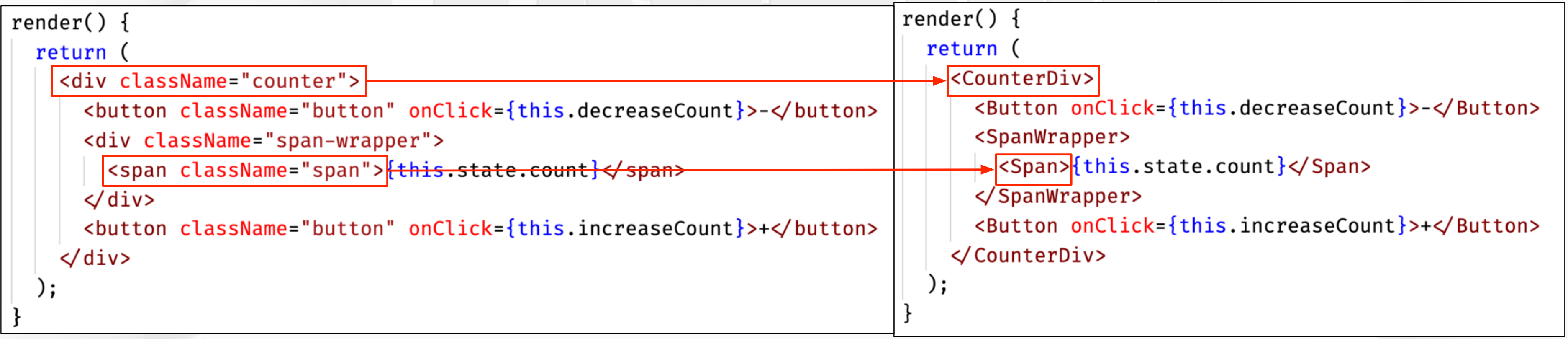
Demo StyledComponent

Fpt Software

2. CSS in JS

Before Styled

After Styled

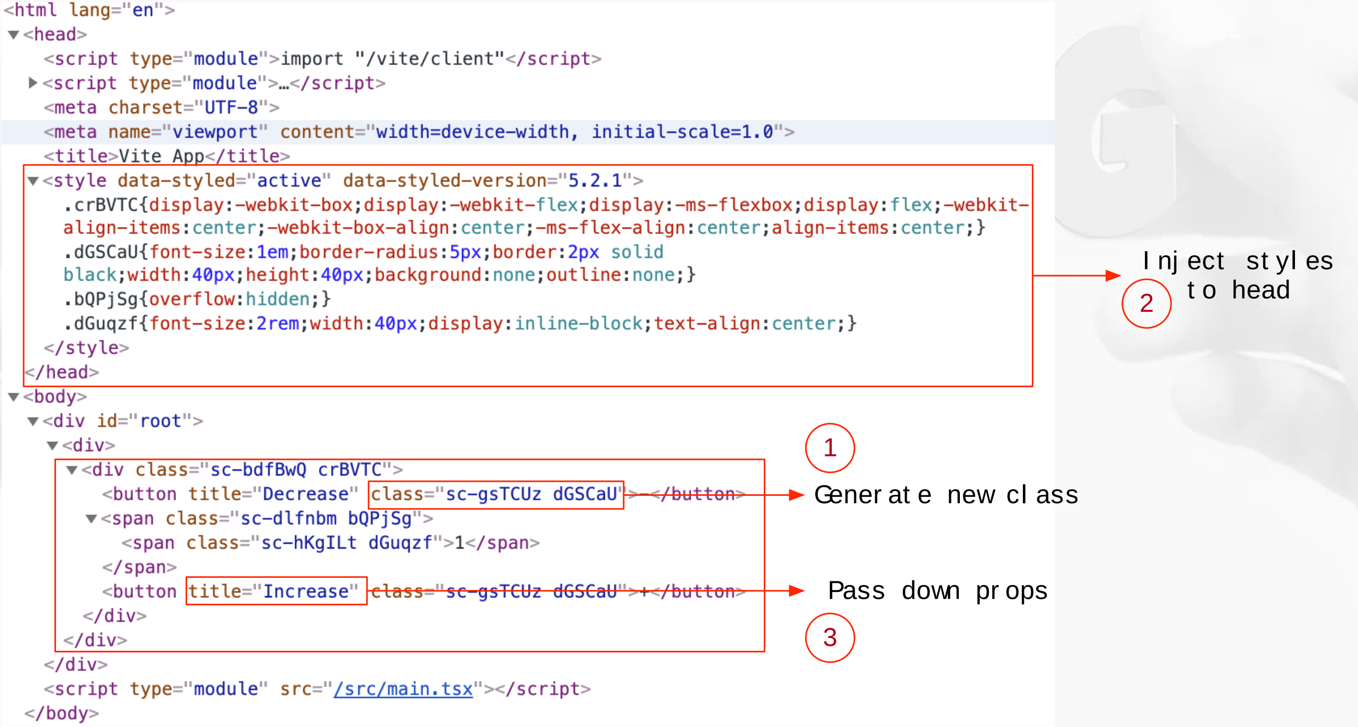


```
render() {
  return (
    <div className="counter">
      <button className="button" onClick={this.decreaseCount}>-</button>
      <div className="span-wrapper">
        <span className="span">{this.state.count}</span>
      </div>
      <button className="button" onClick={this.increaseCount}>+</button>
    </div>
  );
}
```

```
render() {
  return (
    <CounterDiv>
      <Button onClick={this.decreaseCount}>-</Button>
      <SpanWrapper>
        <Span>{this.state.count}</Span>
      </SpanWrapper>
      <Button onClick={this.increaseCount}>+</Button>
    </CounterDiv>
  );
}
```

2. CSS in JS

- What happended under the hood ?



```
<html lang="en">
▼<head>
    <script type="module">import "/vite/client"</script>
  ▶<script type="module">…</script>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Vite App</title>
  ▼<style data-styled="active" data-styled-version="5.2.1">
      .crBVTC{display:-webkit-box;display:-webkit-flex;display:-ms-flexbox;display:flex;-webkit-
      align-items:center;-webkit-box-align:center;-ms-flex-align:center;align-items:center;}
      .dGSCaU{font-size:1em;border-radius:5px;border:2px solid
      black;width:40px;height:40px;background:none;outline:none;}
      .bQPjSg{overflow:hidden;}
      .dGuqzf{font-size:2rem;width:40px;display:inline-block;text-align:center;}
    </style>
  </head>
▼<body>
  ▼<div id="root">
    ▼<div>
      ▼<div class="sc-bdfBwQ crBVTC">
          <button title="Decrease" class="sc-gsTCUz dGSCaU">–</button>
        ▼<span class="sc-dlfnbm bQPjSg">
            <span class="sc-hKgILt dGuqzf">1</span>
          </span>
          <button title="Increase" class="sc-gsTCUz dGSCaU">+</button>
        </div>
      </div>
    </div>
    <script type="module" src="/src/main.tsx"></script>
  </body>
```

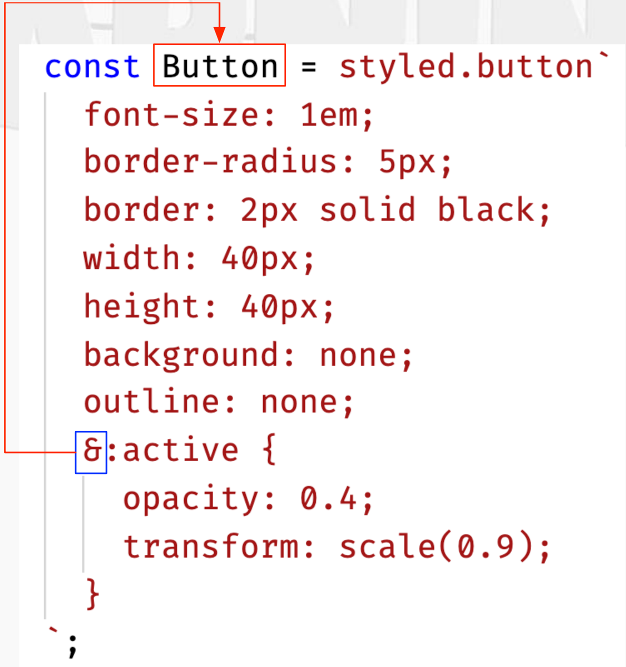
2 Inject styles to head

1 Generate new class

Pass down props 3

2. CSS in JS

- But how do I add style to button when it is clicked ?
  - Use :active pseudo selector

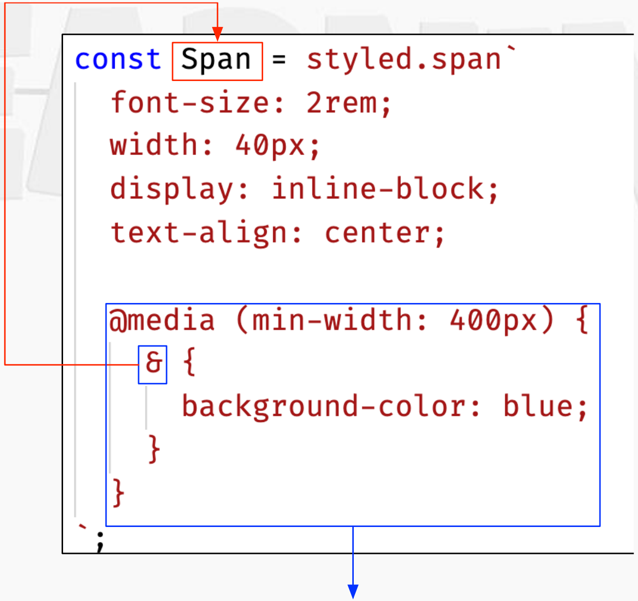


```
const Button = styled.button`
  font-size: 1em;
  border-radius: 5px;
  border: 2px solid black;
  width: 40px;
  height: 40px;
  background: none;
  outline: none;
  &:active {
    opacity: 0.4;
    transform: scale(0.9);
  }
`;
```

Refer to the Component Button

2. CSS in JS

- Does Styled support Responsive styles (i.e @media) ?



```
const Span = styled.span`
  font-size: 2rem;
  width: 40px;
  display: inline-block;
  text-align: center;

  @media (min-width: 400px) {
    & {
      background-color: blue;
    }
  }
`;
```

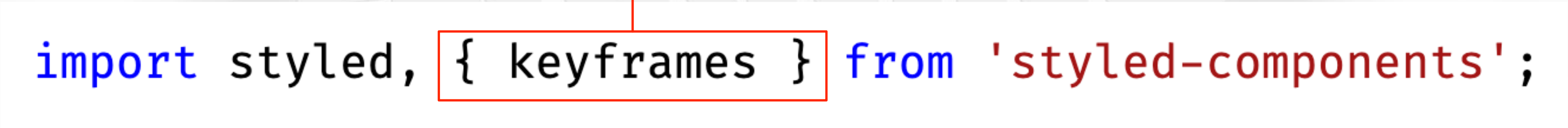
Refer to the Component Span

Apply this style when screen width is greater than 400px

2. CSS in JS

- Can we add animation ?
  - Use keyframes from ‘styled-component’s

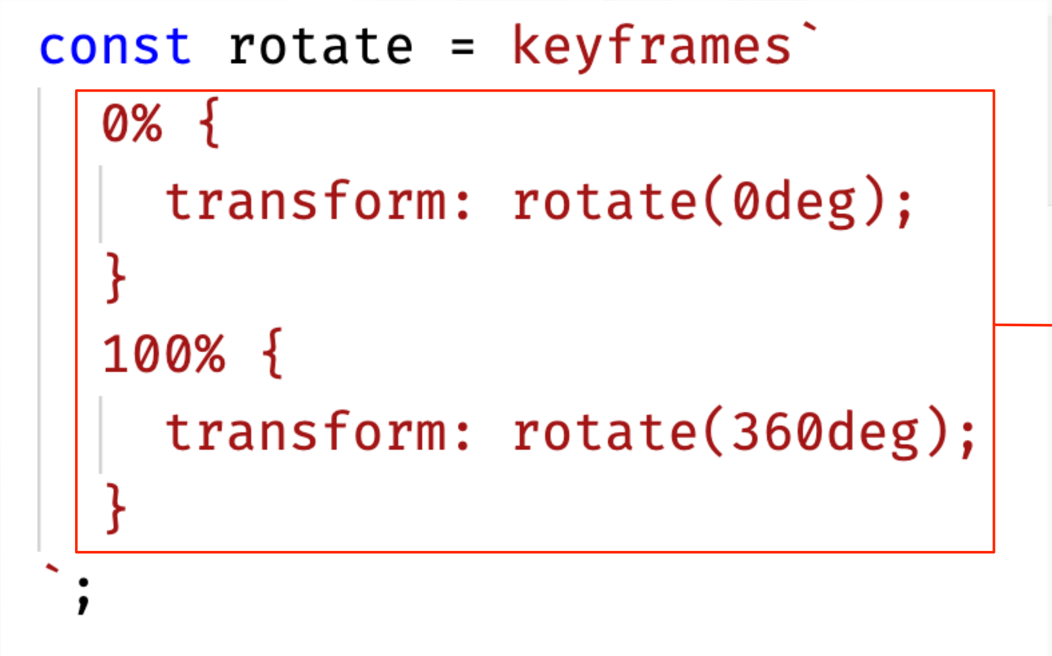
1 Add keyframes utility



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

2. CSS in JS

- Can we add animation ?



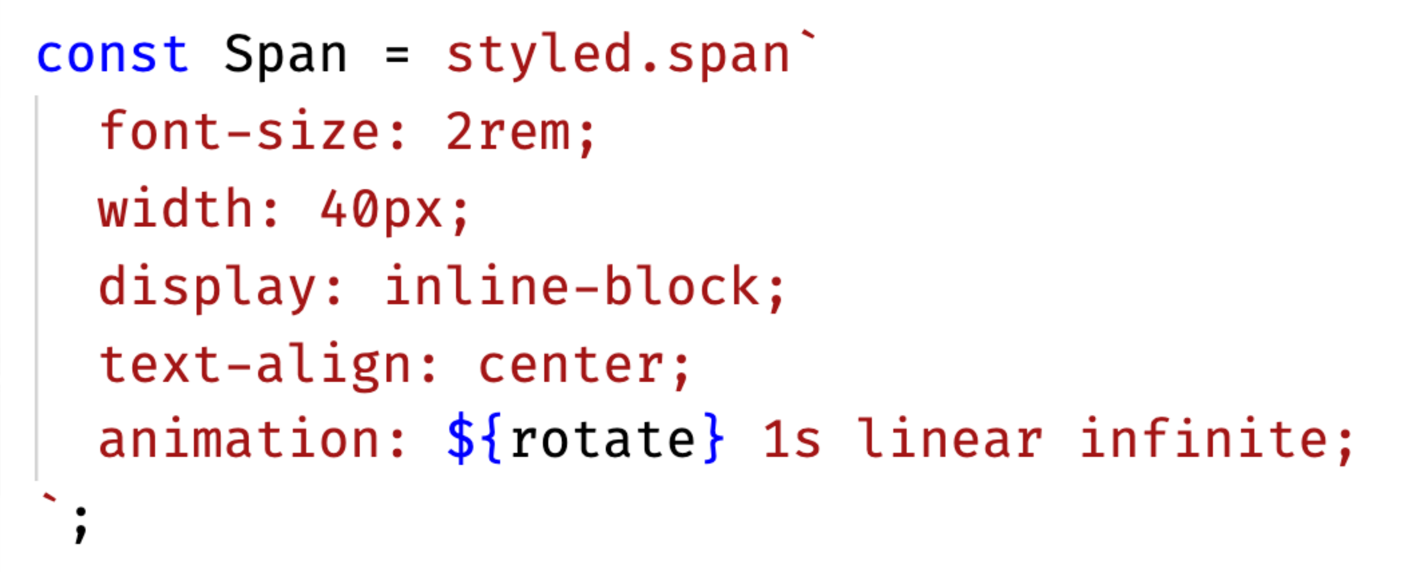
```
const rotate = keyframes`
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
`;
```

2

Create an animation frame with keyframes

2. CSS in JS

- Can we add animation ?

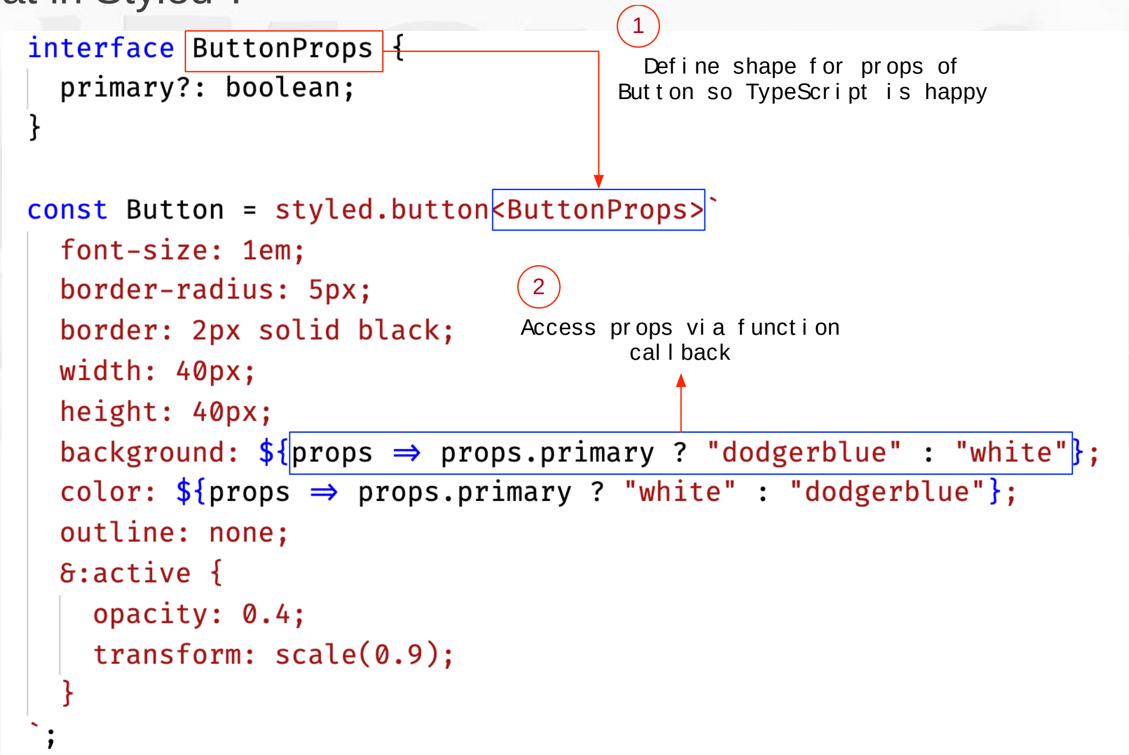


```
const Span = styled.span`
  font-size: 2rem;
  width: 40px;
  display: inline-block;
  text-align: center;
  animation: ${rotate} 1s linear infinite;
`;
```

3 Apply newly created animation frame to Span Component

2. CSS in JS

- Sometimes, we want many variants of a Component (i.e primary Button, secondary, etc…), can we do that in Styled ?



```
interface ButtonProps {
  primary?: boolean;
}

const Button = styled.button<ButtonProps>`
  font-size: 1em;
  border-radius: 5px;
  border: 2px solid black;
  width: 40px;
  height: 40px;
  background: ${props => props.primary ? "dodgerblue" : "white"};
  color: ${props => props.primary ? "white" : "dodgerblue"};
  outline: none;
  &:active {
    opacity: 0.4;
    transform: scale(0.9);
  }
`;
```

1

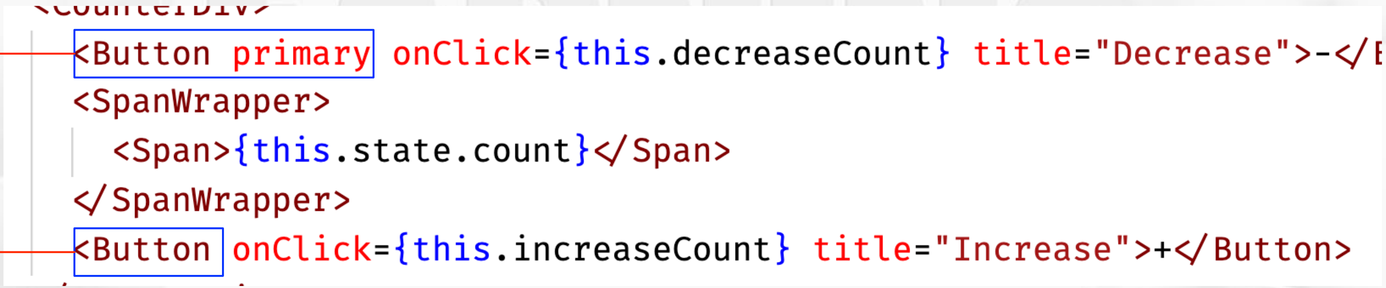
Define shape for props of Button so TypeScript is happy

2

Access props via function callback

2. CSS in JS

- Sometimes, we want many variants of a Component (i.e primary Button, secondary, etc…), can we do that in Styled ?



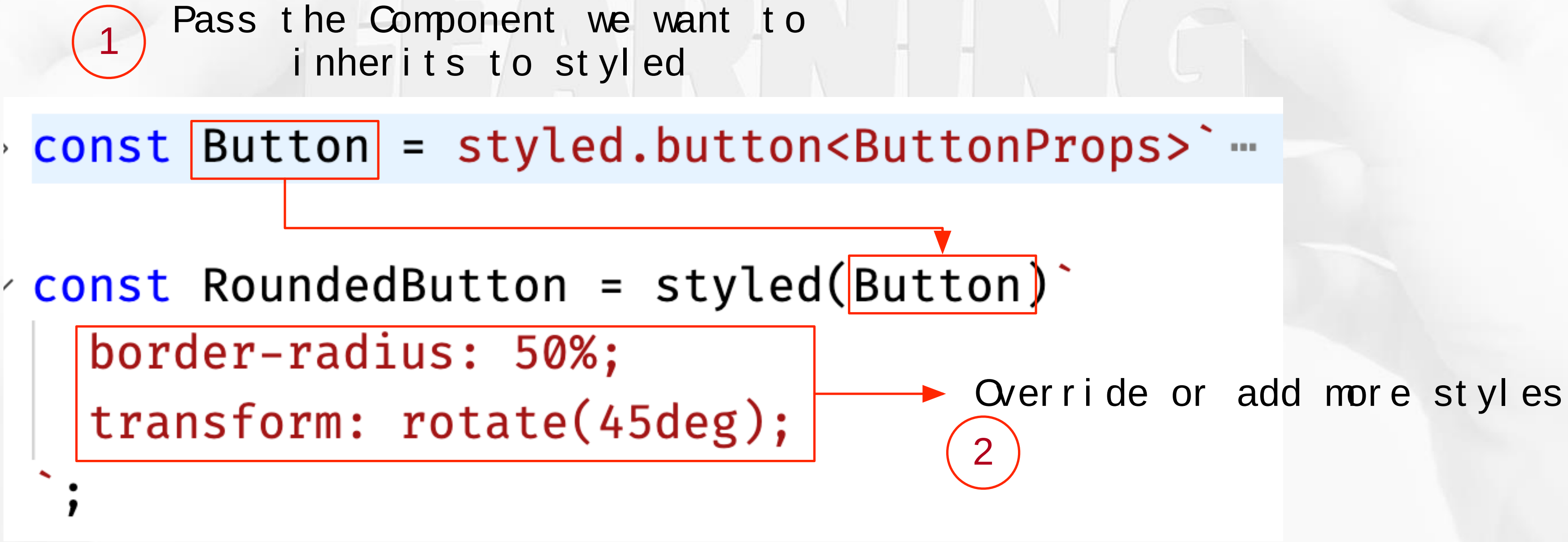
```
<CounterDiv>
  <Button primary onClick={this.decreaseCount} title="Decrease">-</
  <SpanWrapper>
    <Span>{this.state.count}</Span>
  </SpanWrapper>
  <Button onClick={this.increaseCount} title="Increase">+</Button>
```

3

Usage of props

2. CSS in JS

- How do we create new component that inherits the styling of another Styled Component and add more styles ?



1 Pass the Component we want to inherits to styled

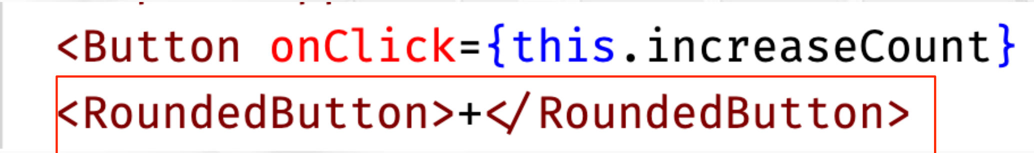
```
const Button = styled.button<ButtonProps>` ...

const RoundedButton = styled(Button)`
  border-radius: 50%;
  transform: rotate(45deg);
`;
```

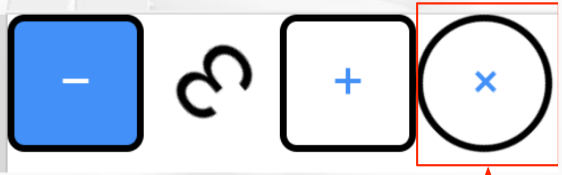
Override or add more styles 2

2. CSS in JS

- How do we create new component that inherits the styling of another Styled Component and add more styles ?



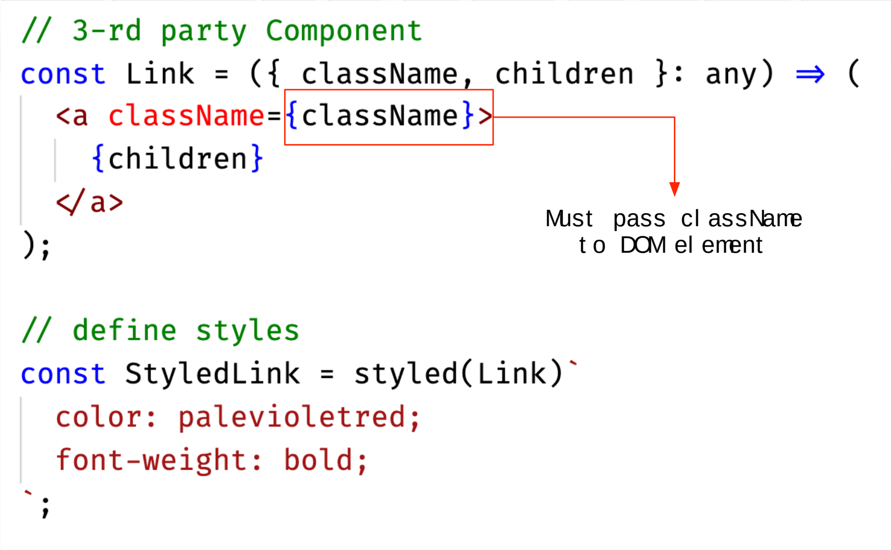
```
<Button onClick={this.increaseCount}
<RoundedButton>+</RoundedButton>
```



3 Usage and display

2. CSS in JS

- Can I inherits/override styles of any Components (not nessessary created from Styled) ?
  - The styled method works perfectly on all of your own or any third-party component, as l ong as they attach the passed **className** prop to a DOM element.



```
// 3-rd party Component
const Link = ({ className, children }: any) => (
  <a className={className}>
    {children}
  </a>
);

// define styles
const StyledLink = styled(Link)`
  color: palevioletred;
  font-weight: bold;
`;
```

Must pass className to DOM element

Practice StyledComponent

Fpt Software

2. Summary

- Styled Components try to solve to Global CSS conflict problem
- With Styled Components, we can add styles to:
  - Native DOM Element
  - Another Styled Components
  - Any React Component (if it pass **className** to DOM element)
- Styled Components support pseudo-selector (via **&**), Responsive query media and animations (via **keyframes**)

3. CSS Module

Fpt Software

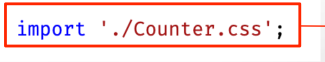
3. CSS Module

- What is CSS Module ?
  - CSS files in which all class names and animation names are scoped **locally** by default.
  - So CSS Modules is not an *official spec* or an *implementation in the browser* but rather a **process in a build step**
- Why should we use CSS Module ?
  1. Live in one place
  2. Only apply to that component and nothing else
- This approach is designed to fix the problem of the global scope in CSS.
- CSS Module are supported in **Create-React-App** and **Vite**

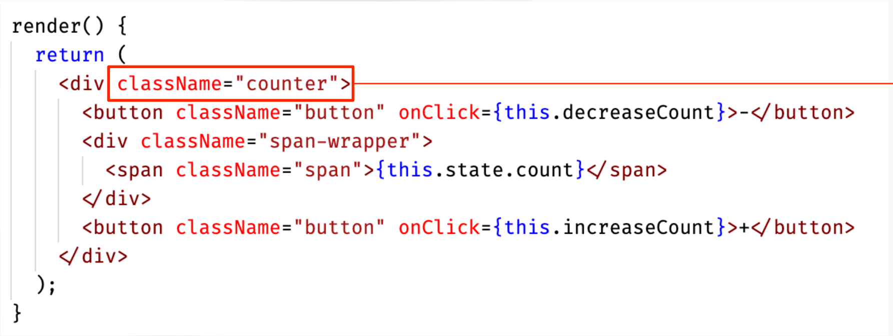
3. CSS Module

- How to use CSS Module ?

Before CSS Module

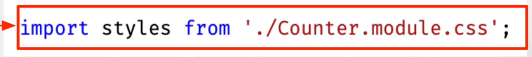


```
import './Counter.css';
```

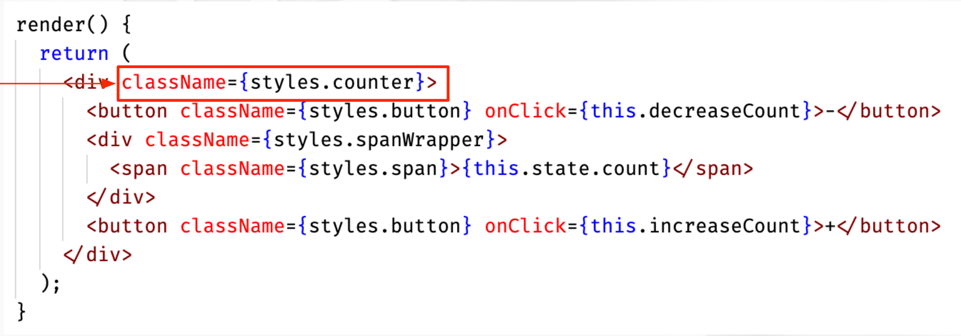


```
render() {
  return (
    <div className="counter">
      <button className="button" onClick={this.decreaseCount}>-</button>
      <div className="span-wrapper">
        <span className="span">{this.state.count}</span>
      </div>
      <button className="button" onClick={this.increaseCount}>+</button>
    </div>
  );
}
```

After CSS Module



```
import styles from './Counter.module.css';
```

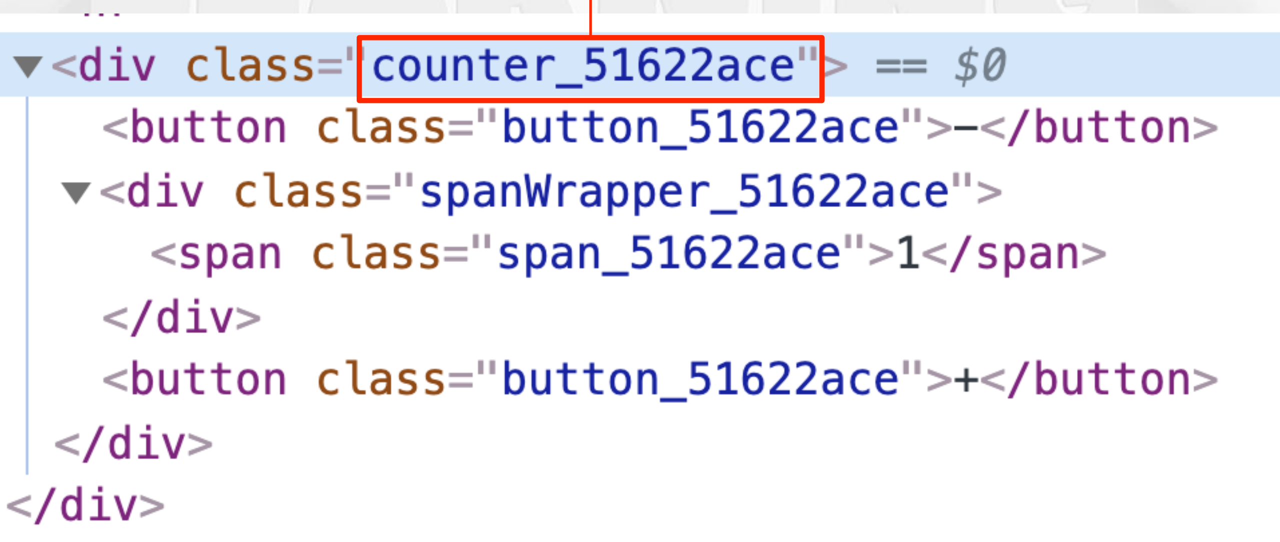


```
render() {
  return (
    <div className={styles.counter}>
      <button className={styles.button} onClick={this.decreaseCount}>-</button>
      <div className={styles.spanWrapper}>
        <span className={styles.span}>{this.state.count}</span>
      </div>
      <button className={styles.button} onClick={this.increaseCount}>+</button>
    </div>
  );
}
```

3. CSS Module

- What happened under the hood ?

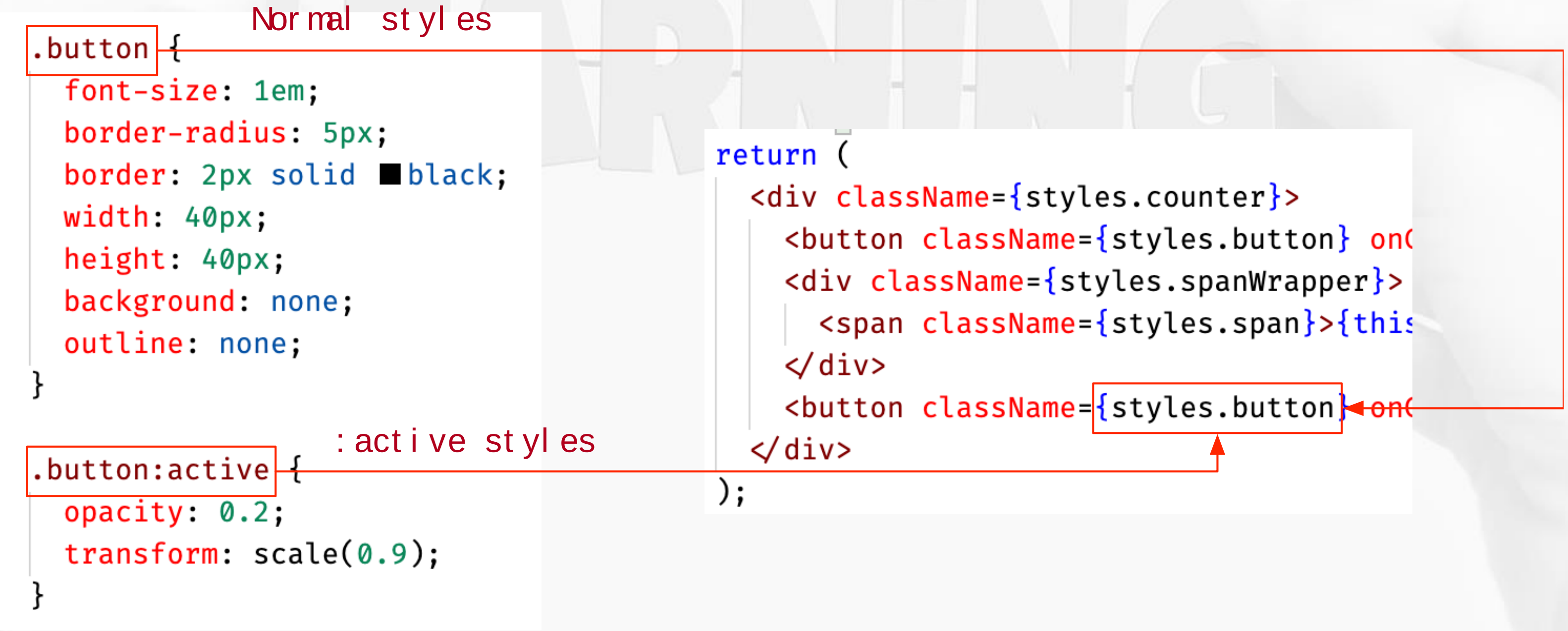
Generate new random classes



```
...
▼<div class="counter_51622ace"> == $0
   <button class="button_51622ace">–</button>
 ▼<div class="spanWrapper_51622ace">
     <span class="span_51622ace">1</span>
   </div>
   <button class="button_51622ace">+</button>
 </div>
</div>
```

3. CSS Module

- About how pseudo selector ?
  - Use .className:active



Normal styles

```css
.button {
  font-size: 1em;
  border-radius: 5px;
  border: 2px solid ■black;
  width: 40px;
  height: 40px;
  background: none;
  outline: none;
}

.button:active {
  opacity: 0.2;
  transform: scale(0.9);
}
```

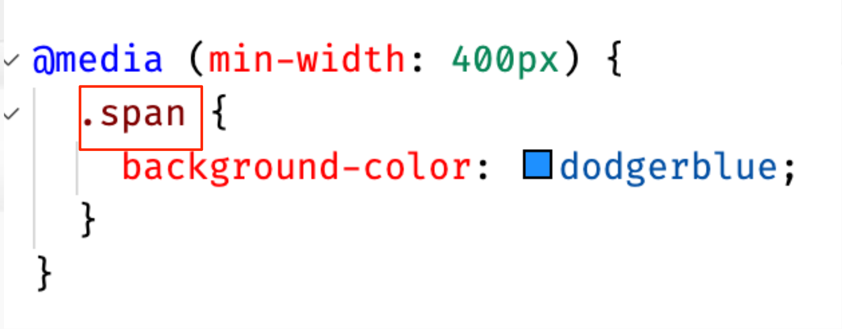
:active styles

```jsx
return (
  <div className={styles.counter}>
    <button className={styles.button} onC
    <div className={styles.spanWrapper}>
      <span className={styles.span}>{this
    </div>
    <button className={styles.button} onC
  </div>
);
```

3. CSS Module

- About how Responsive media query ?

Same like normal CSS Styles but use class

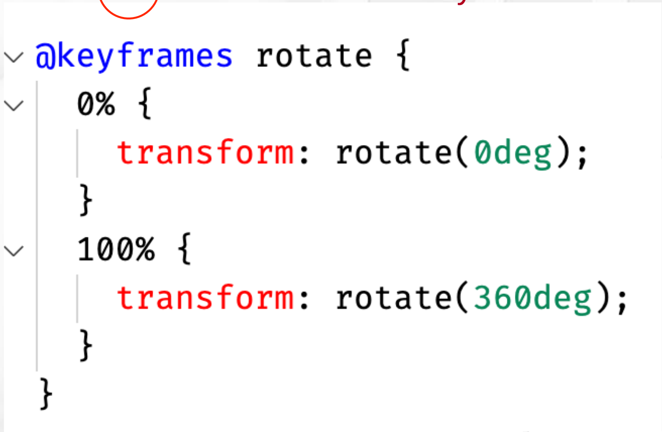


```
@media (min-width: 400px) {
    .span {
      background-color: 🟦dodgerblue;
    }
}
```

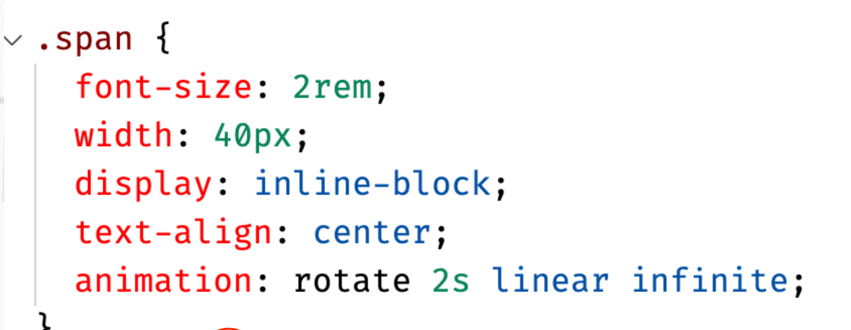
3. CSS Module

- About how about animation ?

1 Define rotate keyframes in CSS

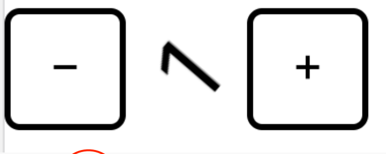


```css
@keyframes rotate {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}
```



```css
.span {
  font-size: 2rem;
  width: 40px;
  display: inline-block;
  text-align: center;
  animation: rotate 2s linear infinite;
}
```

2 Use rotate animation



3 Profit

Practice CSS Module

3. Summary

- CSS Module is designed to fix the problem of the global scope in CSS.
- In CSS Module, CSS files in which all class names and animation names are scoped **locally** by default.
- So CSS Modules is not an *official spec* or an *implementation in the browser* but rather a **process in a build step**
- **Recommended** approach

Happy Coding!

Fpt Software