

---

# Styled Components



What do they do? Do they do stuff? Let's find out!

---

---

# What is it?

- Used to style your React app.
  - Organize your CSS and other Attributes
  - Write styles the “React Way”
  - Extend and Re-Use components
  - Adds functionality to write ‘dynamic’ css
  - Alternative technique compared to using
    - basic css stylesheets
    - inline styles
    - SASS
    - Less
-

---

# Alternative Libraries

Similar, but different

- Emotion - [github.com/tkh44/emotion](https://github.com/tkh44/emotion)
- Styled-jsx - [github.com/zeit/styled-jsx](https://github.com/zeit/styled-jsx)
- Glamorous - [github.com/paypal/glamorous](https://github.com/paypal/glamorous)
- Radium - [github.com/FormidableLabs/radium](https://github.com/FormidableLabs/radium)
- Glamor - [github.com/threepointone/glamor](https://github.com/threepointone/glamor)
- ... And much more.

*<https://alligator.io/react/css-in-js-roundup-styling-react-components/>*

---

# Create a Button

Simple button, vanilla html, no magic yet....

```
<button type="button" name="styled-button">  
|   Simple Button  
</button>
```



Simple Button

# Add Style

Let's make it a blue button!

```
<button  
  type="button"  
  name="styled-button"  
  style={{ color: 'white', backgroundColor: 'blue' }}  
>  
  Simple Button  
</button>
```



Simple Button

# Styled -Componentize!

Let's make our first styled  
component.

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

```
export const MyComponent = styled.div`  
  <CSS GOES IN HERE>  
`;  
;
```

```
export const MyComponent = styled(MocksComponent)`  
  <CSS GOES IN HERE>  
`;  
;
```

```
<MyComponent />
```

# Styled -Componentize!

Let's make our first styled component.

```
import styled from 'styled-components';  
  
export const BlueButton = styled.button`  
  background-color: blue;  
  color: white;  
`;  
;
```

```
import { BlueButton } from './StyledButton.style';
```

```
<BlueButton type="button" name="styled-button">  
  Simple Button  
</BlueButton>
```



Simple Button

# Wait.... What's going on?

- Random string assigned as class name.
- Classname contains styles in global stylesheet
- If styles change, classname is re-named to prevent 'caching'

```
import styled from 'styled-components';  
  
export const BlueButton = styled.button`  
  background-color: blue;  
  color: white;  
`;  
;
```

```
<BlueButton type="button" name="styled-button">  
  Simple Button  
</BlueButton>
```

```
<button class="sc-htpNat kWwXZZ" type=  
"button" name="styled-button">Simple Button  
</button>
```

```
.kWwXZZ {  
  background-color: blue;  
  color: white;  
}
```



# Make it Pretty

Use styled components for other attributes too!

```
export const BlueButton = styled.button.attrs({  
  type: 'button',  
  name: 'styled-button'  
})`  
  background-color: blue;  
  color: white;  
`;  
`;
```

```
<BlueButton>Simple Button</BlueButton>
```



Simple Button

# Extend...

Use `.extend` to create a variation without copy/pasting styles.

```
export const BlueButton = styled.button.attrs({
  type: 'button',
  name: 'styled-button'
})`
  background-color: blue;
  color: white;
`;

export const RoundButton = BlueButton.extend`
  border-radius: 40px;
`;

export const RedButton = RoundButton.extend`
  background-color: red;
  color: black;
`;
```



```
<BlueButton>Simple Button</BlueButton>
<RoundButton>Rounded Button</RoundButton>
<RedButton>Red, Rounded Button</RedButton>
```

# Reusable Animations

Use styled components  
'keyframes' helper to easily  
create re-usable animations.

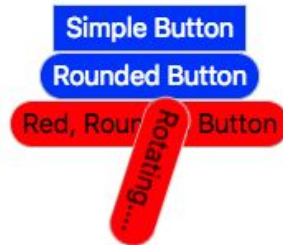
```
import styled, { keyframes } from 'styled-components';

const rotate360 = keyframes`
  from {
    transform: rotate(0deg);
  }

  to {
    transform: rotate(360deg);
  }
`;
```

```
export const RotateButton = styled(RedButton)`
  animation: ${rotate360} 2s linear infinite;
`;
```

```
<RotateButton>Rotating...</RotateButton>
```



# Re-Use the Animation

Use the rotate animation you created for button. On a `<h1>` tag.

```
export const RotatingTitle = styled.h1`  
  animation: ${rotate360} 2s linear infinite;  
`;  
;
```

```
<RotatingTitle>Rotating...</RotatingTitle>
```



# Use Props to make things Dynamic

Like all react components, styled-components have access to props.

```
<MonolithButton rotate="true">Monolith</MonolithButton>
```

```
export const MonolithButton = styled.button.attrs({
  type: 'button',
  name: 'styled-button'
})`
  animation: ${
    (props) => {
      if(props.rotate){
        return `${rotate360} 2s linear infinite`;
      }else{
        return 'none';
      }
    }
  };
`;
```

```
animation: ${({rotate}) => rotate ? `${rotate360} 2s linear infinite` : 'none'};
```

# Make EVERYTHING dynamic

```
<MonolithButton  
  rotate="true"  
  color="white"  
  background="red"  
  round="true"  
>  
  
  Monolith  
</MonolithButton>
```

```
export const MonolithButton = styled.button.attrs({  
  type: 'button',  
  name: 'styled-button'  
})`  
  animation: ${props => props.rotate ? `${rotate360} 2s linear infinite` : `none`};  
  color: ${props => props.color};  
  background-color: ${props => props.background};  
  border-radius ${props => (props.round ? '40px' : '0')};  
`;  
;
```

# Make EVERYTHING dynamic

4 extended components OR  
1 component with 4 props?

Up to you.

```
<BlueButton>Simple Button</BlueButton>  
<RoundButton>Rounded Button</RoundButton>  
<RedButton>Red, Rounded Button</RedButton>
```



```
<Button background="blue" color="white">Simple Button</Button>  
<Button background="blue" color="white" round="true">Simple Button</Button>  
<Button background="red" round="true">Simple Button</Button>
```

# OPINION ALERT

Personally, I like using extend.

```
<PageContainer>
  <TitleRow>{t('This is the title')}</TitleRow>
  <PageRow>
    <DatepickerCol>
      <Datepicker />
    </DatepickerCol>
  </PageRow>
  <VisitorChartRow />
  <PieChartsRow />
  <PageViewsRow />
</PageContainer>
```



---

# ON THE WORLD WIDE WEB



[twitter.com/mackbrowne](https://twitter.com/mackbrowne)



[linkedin.com/in/mackbrowne](https://linkedin.com/in/mackbrowne)



[github.com/mackbrowne](https://github.com/mackbrowne)



[instagram.com/mackbrowne](https://instagram.com/mackbrowne) (Mostly Cats)

---