

React + CSS

Presented by
Tom Isles

Canva



Styling your components

Our components aren't much good if they don't *look* right. Remember, React is a framework for building beautiful user interfaces.

Luckily for us, we can use CSS to style our components, just like we would standard HTML.

Create a new React app

Let's dive straight in! We'll be using **create-react-app** again to create a react application to test out some CSS.

```
npx create-react-app react-css-app
```

App.js

Start by editing your **App.js** file to display the date and time.

```
import * as React from 'react';

function App() {
  const date = new Date(Date.now());
  return (
    <div>
      {date.toLocaleString()}
    </div>
  );
}

export default DateAndTime;
```

Inline Styles

We can use the **style** attribute, which allows us to pass in a javascript object to define our style. Let's try.

```
import * as React from 'react';

function App() {
  const date = new Date(Date.now());
  const style = { color: 'red' };

  return (
    <div style={style}>
      {date.toLocaleString()}
    </div>
  );
}

export default App;
```

style is just a plain javascript object with our styles inside.



Inline Styles

We can use the **style** attribute, which allows us to pass in a javascript object to define our style. Let's try.

```
import * as React from 'react';

function App() {
  const date = new Date(Date.now());
  return (
    <div style={{ color: 'red' }}>{date.toLocaleString()}
```

We can also
make it inline,
like so



Classes

Inline styles have their place, but we can use CSS classes too.

```
import * as React from 'react';

function App() {
  const date = new Date(Date.now());
  return (
    <div className="App">
      {date.toLocaleString()}
    </div>
  );
}

export default App;
```

App.css

```
.App {  
  max-width: 400px;  
  text-align: center;  
  margin: 8px;  
  padding: 32px;  
  border-radius: 4px;  
  background-color: black;  
  color: white;  
  transition: transform 0.25s ease-in-out;  
}  
  
.App:hover {  
  transform: scale(1.01);  
}
```


App.js

Import the CSS file like we would any other file. When the page reloads, the date will have the styles we imported.

```
import * as React from 'react';
import './App.css';

function App() {
  const date = new Date(Date.now());
  return (
    <div className="App">
      {date.toLocaleString()}
    </div>
  );
}

export default App;
```

Combining the two

We can combine class names and inline styles as we see fit. To the right is a more complicated example. Feel free to have a play around and see what you can make!

```
import * as React from 'react';
import './App.css';

const headerStyle = {
  fontSize: '24px',
  padding: '16px',
};

const dateStyle = {
  marginTop: '8px',
};

function App() {
  const date = new Date(Date.now());
  return (
    <div className="App">
      <h1 style={headerStyle}>Date and Time</h1>
      <p style={dateStyle}>{date.toLocaleString()}</p>
    </div>
  );
}
```

Scopes

The great thing about importing CSS in a file is that the CSS can only be used in the file that it is imported in. We shouldn't be able to use css styles from files that we didn't import, right?

```
import * as React from 'react';
import './App.css';

function App() {
  const date = Date.now();
  return (
    <div className="ClassFromAnotherCSSFile">
      {date.toLocaleString()}
    </div>
  );
}

export default App;
```

Wrong. If a CSS file is imported somewhere, you can use its classes anywhere. This way of importing is quite basic.

Scopes

Q

Can we ensure that only the JS files which import a CSS file get to use its classes?

A

Yes! We can use CSS Modules, CSS in JS, and other methods. In later lectures, we will examine some of these methods.

Scoping in this way lets our files stay properly isolated from each other.



Great Job!