Usage

Get started with React and MUI in no time.

MUI components work in isolation. **They are self-supporting**, and will only inject the styles they need to display. They don't rely on any global style-sheets such as normalize.css.

You can use any of the components as demonstrated in the documentation. Please refer to each component's demo page to see how they should be imported.

Quick start

Here's a quick example to get you started, **it's literally all you need**:

```
import * as React from 'react';
import ReactDOM from 'react-dom';
import Button from '@mui/material/Button';

function App() {
  return <Button variant="contained">Hello World</Button>;
}

ReactDOM.render(<App />, document.querySelector('#app'));
```

Yes, this really is all you need to get started, as you can see in this live and interactive demo:

{{"demo": "Usage.js", "hideToolbar": true, "bg": true}}

Globals

MUI usage experience can be improved with a handful of important globals that you'll need to be aware of.

Responsive meta tag

MUI is developed mobile-first, a strategy in which we first write code for mobile devices, and then scale up components as necessary using CSS media queries. To ensure proper rendering and touch zooming for all devices, add the responsive viewport meta tag to your `<head>` element.

```
<meta name="viewport" content="initial-scale=1, width=device-width" />
```

CssBaseline

MUI provides an optional CssBaseline component. It fixes some inconsistencies across browsers and devices while providing slightly more opinionated resets to common HTML elements.

Versioned Documentation

This documentation always reflects the latest stable version of MUI. You can find older versions of the documentation on a separate page.

Next steps

Now that you have an idea of the basic setup, it's time to learn more about:

- How to provide the Material Design font and typography.
- How to take advantage of the theming solution.
- How to override the look and feel of the components.