

CSS Baseline

MUI provides a `CssBaseline` component to kickstart an elegant, consistent, and simple baseline to build upon.

```
{{"component": "modules/components/ComponentLinkHeader.js", "design": false}}
```

Global reset

You might be familiar with [normalize.css](#), a collection of HTML element and attribute style-normalizations.

```
import * as React from 'react';
import CssBaseline from '@mui/material/CssBaseline';

export default function MyApp() {
  return (
    <React.Fragment>
      <CssBaseline />
      {/* The rest of your application */}
    </React.Fragment>
  );
}
```

Scoping on children

However, you might be progressively migrating a website to MUI, using a global reset might not be an option. It's possible to apply the baseline only to the children by using the `ScopedCssBaseline` component.

```
import * as React from 'react';
import ScopedCssBaseline from '@mui/material/ScopedCssBaseline';
import MyApp from './MyApp';

export default function MyApp() {
  return (
    <ScopedCssBaseline>
      {/* The rest of your application */}
      <MyApp />
    </ScopedCssBaseline>
  );
}
```

⚠ Make sure you import `ScopedCssBaseline` first to avoid box-sizing conflicts as in the above example.

Approach

Page

The `<html>` and `<body>` elements are updated to provide better page-wide defaults. More specifically:

- The margin in all browsers is removed.

- The default Material Design background color is applied. It's using `theme.palette.background.default` for standard devices and a white background for print devices.
- If `enableColorScheme` is provided to `CssBaseline`, native components color will be set by applying `color-scheme` on `<html>`. The value used is provided by the theme property `theme.palette.mode`.

Layout

- `box-sizing` is set globally on the `<html>` element to `border-box`. Every element—including `::before` and `::after` are declared to inherit this property, which ensures that the declared width of the element is never exceeded due to padding or border.

Scrollbars

This API is deprecated, consider using [color-scheme](#) instead.

The colors of the scrollbars can be customized to improve the contrast (especially on Windows). Add this code to your theme (for dark mode).

```
import darkScrollbar from '@mui/material/darkScrollbar';
const theme = createTheme({
  components: {
    MuiCssBaseline: {
      styleOverrides: {
        body: theme.palette.mode === 'dark' ? darkScrollbar() : null,
      },
    },
  },
});
```

Be aware, however, that using this utility (and customizing `-webkit-scrollbar`) forces MacOS to always show the scrollbar.

Color scheme

This API is introduced in @mui/material (v5.1.0) for switching between "light" and "dark" modes of native components such as scrollbar, using the `color-scheme` CSS property. To enable it, you can set `enableColorScheme=true` as follows:

```
<CssBaseline enableColorScheme />

// or

<ScopedCssBaseline enableColorScheme >
  {/* The rest of your application using color-scheme*/}
</ScopedCssBaseline>
```

Typography

- No base font-size is declared on the `<html>`, but 16px is assumed (the browser default). You can learn more about the implications of changing the `<html>` default font size in [the theme documentation](#) page.
- Set the `theme.typography.body1` style on the `<body>` element.

- Set the font-weight to `theme.typography.fontWeightBold` for the `` and `` elements.
- Custom font-smoothing is enabled for better display of the Roboto font.

Customization

Head to the [global customization](#) section of the documentation to change the output of these components.