Importing and Using Components in MDX

You can import your components from other third party libraries, like `theme-ui`. Use cases for external libraries could be charting libraries for adding rich data visualizations, form components for adding email signups, styled portions of content like pullquotes, or call to action buttons throughout your pages. You can also import and reuse *your own* React components and MDX documents.

Import components for use from another library

Components imported from other libraries can be rendered inline with your markdown content, allowing you to include rich media like charts, interactive buttons, or styled messages. Components are imported at the top of your MDX documents—in the same syntax they are imported in JavaScript files—and then added using opening and closing brackets like normal JSX elements.

To include a component from another library (this example uses the message component from `theme-ui`), you need to import it at the top of your MDX file:

```
---
title: Importing Components Example
---

import { Message } from "theme-ui" // highlight-line

You can import your own components.

<Message>MDX gives you JSX in Markdown!</Message> // highlight-line
```

Note: steps for importing custom components or MDX documents from a relative location in your project are also covered in the Writing Pages in MDX guide.

Make components available globally as shortcodes

To avoid having to import the same component inside of every MDX document you author, you can add components to an `MDXProvider` to make them globally available in MDX pages. This pattern is sometimes referred to as shortcodes.

```
import React from "react"
// highlight-start
import { MDXProvider } from "@mdx-js/react"
import { Chart, Pullquote } from "./ui"
import { Message } from "theme-ui"
// highlight-end

const shortcodes = { Chart, Pullquote, Message } // highlight-line

export default function Layout({ children }) {
  return (
    <MDXProvider components={shortcodes}>{children}</MDXProvider> // highlight-line
  )
}
```

All MDX components passed into the `components` prop of the `MDXProvider` will be made available to MDX documents that are nested under the provider. The `MDXProvider` in this example is in a layout component that wraps all MDX pages, you can read about this pattern in the layout section of the `gatsby-plugin-mdx` README.

Now, you can include components in your MDX without importing them:

```
---
title: Shortcode Components Example
---

Now, if you want to include the Message component, it's available in all MDX documents!

<Message>MDX gives you JSX in Markdown!</Message> // highlight-line

The Chart is also available since it was passed into the MDXProvider:

<Chart /> // highlight-line
```

Because the `<Message />` and `<Chart />` components were passed into the provider, they are available for use in all MDX documents.

Lazy-loading components

When you use components in your `.mdx` files, Gatsby will bundle them into the main application bundle. This can cause performance problems.

In the future, `gatsby-plugin-mdx` will address this. In the meantime, it can be prudent to lazy-load very large dependencies. The following snippet provides an example for lazy-loading an imaginary `Thing` component:

```
import React from "react"
```

```
const Placeholder = () => null

const LazyThing = props => {
  // While the component is loading, we'll render a fallback placeholder.
  // (The Placeholder is a component that renders nothing).
  const [Component, setComponent] = React.useState(() => Placeholder)

  // After the initial render, kick off a dynamic import to fetch
  // the real component, and set it into our state.
  React.useEffect(() => {
    import("./Thing.js").then(Thing => setComponent(() => Thing.default))
  }, [])

  return <Component {...props} />
}

export default LazyThing
```

Inside your MDX, swap out any references to `Thing` with `LazyThing`:

```
-import Thing from "../components/Thing/Thing.js"
+import LazyThing from "../components/Thing/LazyThing.js"

## Introducing Things

Here is a demo:

-<Thing hi={5} />
+<LazyThing hi={5} />
```

Additional resources

- Follow this detailed example on using MDX to import and render components.