

gatsby-plugin-glamor

Provide drop-in support for using the css-in-js library [Glamor](#) including optimized server rendering.

In addition, you can also use this plugin to make [glamorous](#) 🚀 work with server side rendering and start writing React components that carry their styles with them.

Install

```
npm install gatsby-plugin-glamor glamor
```

How to use

First add the plugin to your `gatsby-config.js`.

```
plugins: [`gatsby-plugin-glamor`]
```

Glamor [provides many convenient ways to style your components](#). One particularly convenient (and suggested) way is to use its `css` prop. It works exactly the same as the [default style prop](#) except it supports the entire CSS language. So things not supported by inline styles are supported with Glamor like pseudo-classes/-elements, `@media` queries, parent/child/contextual selectors, etc.

```
render () {
  return (
    <div
      css={{
        margin: `0 auto`,
        border: `1px solid gray`,
      }}
    >
      <h1
        css={{
          color: `red`,
          // Pseudo styles are supported!
          ':hover': {
            textDecoration: `underline`,
          },
          // As are media queries!
          '@media (min-width: 400px)': {
            color: `blue`,
          },
        }}
      >
        This is the title!
      </h1>
      <div>
        The body!
      </div>
    </div>
  )
}
```

```
)  
}
```

The `css` prop works on both the default DOM components as well as most custom components. Under the hood, Glamor converts the `css` prop to a `className` prop so the `css` prop will work as long as your or the 3rd party component you're using uses `className` .