In this guide, you'll learn Gatsby's approach to layouts, how to create and use layout components, and how to prevent layout components from unmounting.

Gatsby's approach to layouts

Gatsby does not, by default, automatically apply layouts to pages (there are, however, ways to do so which will be covered in a later section). Instead, Gatsby follows React's compositional model of importing and using components. This makes it possible to create multiple levels of layouts, e.g. a global header and footer, and then on some pages, a sidebar menu. It also makes it possible to pass data between layout and page components.

What are layout components?

Layout components are for sections of your site that you want to share across multiple pages. For example, Gatsby sites will commonly have a layout component with a shared header and footer. Other common things to add to layouts are a sidebar and/or navigation menu. On this page for example, the header at the top is part of gatsbyjs.com’s layout component.

How to create layout components

It is recommended to create your layout components alongside the rest of your components (e.g. into `src/components/`).

Here is an example of a very basic layout component at `src/components/layout.js` :

```
import React from "react"

export default function Layout({ children }) {
  return (
    <div style={{ margin: `0 auto`, maxWidth: 650, padding: `0 1rem` }}>
      {children}
    </div>
  )
}
```

How to import and add layout components to pages

If you want to apply a layout to a page, you will need to include the `Layout` component and wrap your page in it. For example, here is how you would apply your layout to the front page:

```
import React from "react"
import Layout from "../components/layout" // highlight-line

export default function Home() {
  return (
    <Layout> {/* highlight-line */}
      <h1>I’m in a layout!</h1>
    </Layout> {/* highlight-line */}
  );
}
```

Repeat for every page and template that needs this layout.

How to prevent layout components from unmounting

As mentioned earlier, Gatsby does not, by default, automatically wrap pages in a layout component. The "top level" component is the page itself. As a result, when the "top level" component changes between pages, React will re-render all children. This means that shared components like navigations will unmount and remount. This will break CSS transitions or React state within those shared components.

If you need to set a wrapper component around page components that won't get unmounted on page changes, use the **`wrapPageElement`** browser API and the SSR equivalent.

Alternatively, you can prevent your layout component from unmounting by using gatsby-plugin-layout, which implements the `wrapPageElement` APIs for you.

Other resources

- Creating nested layout components in Gatsby
- Life after layouts in Gatsby V2
- Migrating from v1 to v2
- gatsby-plugin-layout
- wrapPageElement Browser API
- wrapPageElement SSR API