

## Styling in Next.js

13 min

A Next.js application can be styled in several different ways. By default, Next.js has built-in support for Global CSS, CSS

Preview: Docs Loading link description

### [Modules](#)

, [Tailwind CSS](#), [CSS-in-JS](#), and [Sass](#).

In this exercise, we will use CSS Modules and Global CSS.

CSS Modules is a great way to prevent style name collision by modularizing CSS files. Files will be processed as a CSS Module if appended with the `.module.css` extension. Location-wise, CSS Module files can be colocated with the component they are styling.

For example:

```
import styles from './HomePage.modules.css'

export default function Page() {
  return <h1 className={styles.header}>Hello, World!</h1>
}
```

Global CSS files can apply a style to the entire application. The global CSS file can be imported into any page or layout. A layout is a special component that will be detailed in the upcoming Routing lesson, but typically, the global CSS file is imported into the *root* layout, a layout file that sits in the `/app` directory.

For example:

```
//layout.tsx
import './global.css'

export default function RootLayout({
  children,
}: {
  children: React.ReactNode
}) {
  return (
    <html lang="en">
      <body>{children}</body>
    </html>
  )
}
```

```
}
```

## Instructions

1. Checkpoint 1 Passed

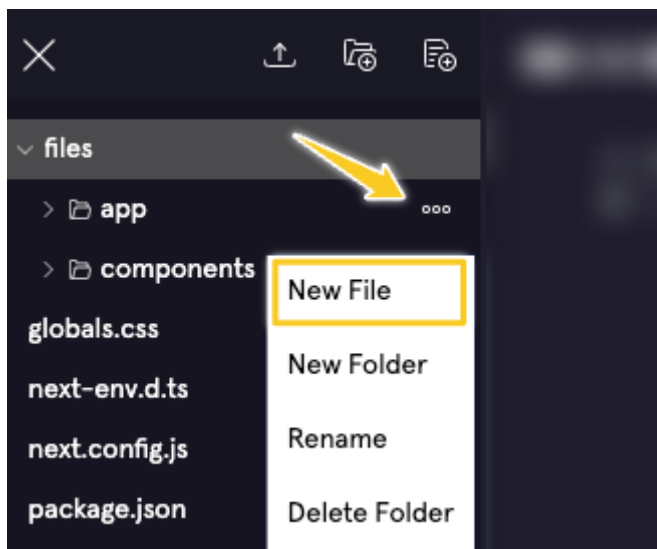
### 1.

In the `/app` directory, create a CSS Module with the name `HomePage`. In the stylesheet, define a style for the `intro` class name.

Hint

The file navigation pane is on the top left of the code editor.

Reference the following to add a new file to the `/app` directory:



Ensure your `HomePage.module.css` file is *in* the `/app` directory.

2. Checkpoint 2 Passed

### 2.

Return to `page.tsx`. Import the newly defined CSS Module file into a `styles` object, and apply it with the `className` attribute.

Hint

Use a named import to import the CSS Module. To apply a style, the syntax is:

```
<div className={importedStyle.styleName}>
```

3. Checkpoint 3 Passed

### 3.

It's time to apply some global styles. In `globals.css` stylesheet, add an additional style to change the background color for the `<body>` tag.

Hint

Add your style under the existing styling:

```
body {  
  padding: 0;  
  margin: 0;  
  // add your style here.  
}
```

#### 4. Checkpoint 4 Passed

#### 4.

View the applied styles running the code using `npm run dev` in the terminal.

Once you see a console message indicating "Ready," click the Refresh Browser button in the workspace browser bar.

Hint

Enter the command inside the terminal window.

The window can be refreshed when it displays "Ready."

```
> dev
```

```
> next dev -p 4001
```

```
▲ Next.js 14.0.1
```

```
- Local:    http://localhost:4001
```

```
✓ Ready in 1156ms
```

#### **HomePage.module.css**

```
.intro {  
  
  font-size: 2em;  
  
}
```

## **globals.css**

```
body {  
  padding: 0;  
  margin: 0;  
  background-color: #b1cbbb;  
}
```