# gatsby-plugin-postcss

Provides drop-in support for PostCSS

## Install

```
npm install postcss gatsby-plugin-postcss
```

## How to use

1. Include the plugin in your `gatsby-config.js` file.
2. Write your stylesheets using PostCSS (`.css` files) and require or import them as normal.

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

If you need to pass options to PostCSS use the plugins options; see `postcss-loader` for all available options.

If you need to override the default options passed into `css-loader`. **Note:** Gatsby is using `css-loader@^5.0.0`.

```
plugins: [
  {
    resolve: `gatsby-plugin-postcss`,
    options: {
      cssLoaderOptions: {
        camelCase: false,
      },
    },
  },
]
```

### With CSS Modules

Using CSS modules requires no additional configuration. Simply prepend `.module` to the extension. For example: `app.css` -> `app.module.css`. Any file with the `module` extension will use CSS modules. CSS modules are imported as ES Modules to support treeshaking. You'll need to import styles as: `import { yourClassName, anotherClassName } from './app.module.css'`

### PostCSS plugins

If you would prefer to add additional postprocessing to your PostCSS output you can specify plugins in the plugin options:

```
plugins: [
  {
    resolve: `gatsby-plugin-postcss`,
    options: {
      postCssPlugins: [require(`postcss-preset-env`)({ stage: 0 })],
    },
  },
],
```

Alternatively, you can use `postcss.config.js` to specify your particular PostCSS configuration:

```
const postcssPresetEnv = require(`postcss-preset-env`)

module.exports = () => ({
  plugins: [
    postcssPresetEnv({
      stage: 0,
    }),
  ],
})
```

If you need to override the default options passed into `css-loader`.

In this example `css-loader` is configured to output classnames as is, instead of converting them to camel case. Named exports must be disabled for this to work, and so you have to import CSS using `import styles from './file.css` instead of `import * as styles from './file.module.css'`

```
plugins: [
  {
    resolve: `gatsby-plugin-postcss`,
    options: {
      cssLoaderOptions: {
        exportLocalsConvention: false,
        namedExport: false,
      },
    },
  },
]
```