

Gatsby can load plugins from your website's local plugins folder, which is a folder named `plugins` in the website's root directory.

Consider this example project structure which includes a local plugin called `gatsby-local-plugin`:

```
/my-gatsby-site
├── /src
│   ├── /pages
│   └── /components
<!-- highlight-start -->
├── /plugins
│   └── /gatsby-local-plugin
│       ├── /package.json
│       └── /gatsby-node.js
<!-- highlight-end -->
├── gatsby-config.js
├── gatsby-node.js
└── package.json
```

Like the name of the plugins folder implies, you can include multiple plugins in your local plugin folder.

Including a local plugin in your plugins folder also requires a configuration step (similar to a third-party plugin you've installed in your `node_modules` folder by running `npm install`); just as plugins installed from npm need to be included in your `gatsby-config`, you need to add the name of your local plugin to the plugins array as well:

```
module.exports = {
  plugins: [
    `gatsby-third-party-plugin`,
    `gatsby-local-plugin`, // highlight-line
  ],
}
```

Verifying your plugin is loading

To verify that your plugin is available for use in your Gatsby site, you can add a small snippet of code to a `gatsby-node.js` file (you may need to add the `gatsby-node.js` file if there isn't one already) in the root of your plugin:

```
exports.onPreInit = () => {
  console.log("Testing...")
}
```

The [`onPreInit` API](#) is the first Node API called by Gatsby right after plugins are loaded.

Then, when running your site in develop or build mode, you should see "Testing..." logged in your terminal:

```
success open and validate gatsby-configs - 0.051s
success load plugins - 1.047s
Testing... // highlight-line
success onPreInit - 0.023s
...
```

Loading local plugins from *outside* the plugins folder

If you want to reference a plugin that is not in the plugins folder, there are several options that are described in more detail in the [Creating a Local Plugin guide](#).