

Learn what the *Yarn* package manager is, how to use it, and how it fits in to the Gatsby ecosystem.

What is Yarn?

Yarn is a package manager for the [Node.js](#) JavaScript runtime. It's an alternative to Node's standard package manager, [npm](#). [Contributing](#) to Gatsby core requires Yarn. Gatsby core uses Yarn's [workspaces](#) feature to manage dependencies. Gatsby [theme development](#) also uses Yarn workspaces. For Gatsby site development, you can use Yarn or npm.

Note: Most tutorials and site development examples from the Gatsby docs use *npm*. To avoid confusion and possible conflicts with your dependencies, Gatsby recommends using *npm* for site development.

Engineers from Facebook, Google, Exponent, and Tilde [launched Yarn](#) in 2016 to improve package management for large-scale, monolithic repositories. A monolithic repository, or [monorepo](#), contains the code for many different projects in a single repository. Gatsby core uses a monorepo pattern for its code.

As a monorepo adds projects and contributors, the number and size of its dependencies also increases. Yarn mitigates this in two ways.

1. It caches previously downloaded packages.
2. It resolves duplicate dependencies so that a package is only downloaded once.

As a result, installing or updating a monorepo often takes less time with Yarn than with npm.

Installing Yarn

You'll need to install Yarn separately from Node. Choose the [binary package](#) for your operating system. For Gatsby core development, you'll need Yarn version 1.0.2 or later.

Although the Yarn documentation discourages it, you can also install Yarn using npm or npx. Use `npm install yarn` with the `-g` or `--global` flags to install Yarn globally. Or install Yarn per project using npx: `npx yarn`. If you install Yarn using npx, you'll need to prefix Yarn commands with `npx`, e.g. `npx yarn add`. Read more about [npm](#) and npx in the Gatsby docs.

Using Yarn to install Gatsby

Once installed, you can use Yarn to install the Gatsby CLI globally or locally. If you'd like to use Gatsby CLI commands such as `gatsby new`, install the CLI globally using the [global](#) [prefix](#).

```
yarn global add gatsby-cli
```

You can also install Gatsby locally using `yarn add gatsby-cli`. If you use this method, you'll need to prefix Gatsby commands with `yarn`, for example, `yarn gatsby develop`.

Note: If you've installed Yarn with npx, you'll need to use both the *npx* and *yarn* prefixes, e.g.: `npx yarn add gatsby-cli`. Yarn will ignore the `global` prefix if you've installed it using npx.

Using Yarn as your Gatsby package manager

When you run `gatsby new` for the first time, you'll be prompted to choose npm or Yarn as your default package manager. You can choose Yarn then, or change it at a later time. To switch from npm to Yarn, edit the Gatsby CLI configuration file available at `~/.config/gatsby/config.json` to add the following lines.

```
{
  "cli": {
    "packageManager": "yarn"
  }
}
```

Gatsby will use Yarn as the package manager for new projects. You can also install Gatsby plugins using Yarn. Replace `npm install` or `npm i` with `yarn add`.

Using Yarn to install packages generates a `yarn.lock` file. The `yarn.lock` tracks the exact version that you installed with `yarn add`, similar to `package-lock.json`. Commit `yarn.lock` to your project's repository. Doing so keeps your dependencies consistent across team members and computers.

Learn more about Yarn

- [The package.json guide](#) from Nodejs.dev
- [Yarn](#) official website
- [Building a Theme](#)
- [Gatsby Theme Authoring](#) from egghead.io
- [How to change your default package manager for your next project?](#) from the Gatsby CLI docs
- [Setting Up Your Local Dev Environment](#) for contributions to the Gatsby project