**Create a Next.js App**

[**1**](https://nextjs.org/learn/basics/create-nextjs-app)

[2](https://nextjs.org/learn/basics/create-nextjs-app/setup)

[3](https://nextjs.org/learn/basics/create-nextjs-app/welcome-to-nextjs)

[4](https://nextjs.org/learn/basics/create-nextjs-app/editing-the-page)

To build a complete web application with React from scratch, there are many important details you need to consider:

* Code has to be bundled using a bundler like webpack and transformed using a compiler like Babel.
* You need to do production optimizations such as code splitting.
* You might want to statically pre-render some pages for performance and SEO. You might also want to use server-side rendering or client-side rendering.
* You might have to write some server-side code to connect your React app to your data store.

A *framework* can solve these problems. But such a framework must have the right level of abstraction — otherwise it won’t be very useful. It also needs to have great "Developer Experience", ensuring you and your team have an amazing experience while writing code.

**Next.js: The React Framework**

Enter **Next.js**, the React Framework. Next.js provides a solution to all of the above problems. But more importantly, it puts you and your team in the [pit of success](https://blog.codinghorror.com/falling-into-the-pit-of-success/) when building React applications.

Next.js aims to have best-in-class developer experience and many built-in features, such as:

* An intuitive [page-based](https://nextjs.org/docs/basic-features/pages) routing system (with support for [dynamic routes](https://nextjs.org/docs/routing/dynamic-routes))
* [Pre-rendering](https://nextjs.org/docs/basic-features/pages#pre-rendering), both [static generation](https://nextjs.org/docs/basic-features/pages#static-generation-recommended) (SSG) and [server-side rendering](https://nextjs.org/docs/basic-features/pages#server-side-rendering) (SSR) are supported on a per-page basis
* Automatic code splitting for faster page loads
* [Client-side routing](https://nextjs.org/docs/routing/introduction#linking-between-pages) with optimized prefetching
* [Built-in CSS](https://nextjs.org/docs/basic-features/built-in-css-support) and [Sass support](https://nextjs.org/docs/basic-features/built-in-css-support#sass-support), and support for any [CSS-in-JS](https://nextjs.org/docs/basic-features/built-in-css-support#css-in-js) library
* Development environment with [Fast Refresh](https://nextjs.org/docs/basic-features/fast-refresh) support
* [API routes](https://nextjs.org/docs/api-routes/introduction) to build API endpoints with Serverless Functions
* Fully extendable

Next.js is used in tens of thousands of production-facing websites and web applications, including many of the world's largest brands.

**About This Tutorial**

This free interactive course will guide you through how to get started with Next.js.

In this tutorial, you’ll learn Next.js basics by creating a very simple **blog app**. Here’s an example of the final result:

[**https://next-learn-starter.vercel.app**](https://next-learn-starter.vercel.app/) ([source](https://github.com/vercel/next-learn/tree/master/basics/demo))

This tutorial assumes basic knowledge of JavaScript and React. If you’ve never written React code, you should go through [the official React tutorial](https://reactjs.org/tutorial/tutorial.html) first.

If you’re looking for documentation instead, [visit the Next.js documentation](https://nextjs.org/docs/getting-started).

**Join the Conversation**

If you have questions about anything related to Next.js or this course, you're welcome to ask our community on [Discord](https://discord.gg/Q3AsD4efFC).

Let’s get started!

**Create a Next.js App**

[1](https://nextjs.org/learn/basics/create-nextjs-app)

[**2**](https://nextjs.org/learn/basics/create-nextjs-app/setup)

[3](https://nextjs.org/learn/basics/create-nextjs-app/welcome-to-nextjs)

[4](https://nextjs.org/learn/basics/create-nextjs-app/editing-the-page)

**Setup**

First, let’s make sure that your development environment is ready.

* If you don’t have **Node.js** installed, [install it from here](https://nodejs.org/en/). You’ll need Node.js version **10.13** or later.
* You’ll be using your own text editor and terminal app for this tutorial.

If you are on Windows, we recommend [downloading Git for Windows](https://gitforwindows.org/) and use Git Bash that comes with it, which supports the UNIX-specific commands in this tutorial. [Windows Subsystem for Linux (WSL)](https://docs.microsoft.com/en-us/windows/wsl/install-win10) is another option.

**Create a Next.js app**

To create a Next.js app, open your terminal, cd into the directory you’d like to create the app in, and run the following command:

npx create-next-app@latest nextjs-blog --use-npm --example "https://github.com/vercel/next-learn/tree/master/basics/learn-starter"

Under the hood, this uses the tool called [create-next-app](https://nextjs.org/docs/api-reference/create-next-app), which bootstraps a Next.js app for you. It uses [this template](https://github.com/vercel/next-learn/tree/master/basics/learn-starter) through the --example flag.

If it doesn’t work, please take a look at [this page](https://github.com/vercel/next-learn/blob/master/basics/errors/install.md).

**Run the development server**

You now have a new directory called nextjs-blog. Let’s cd into it:

cd nextjs-blog

Then, run the following command:

npm run dev

This starts your Next.js app’s "development server" (more on this later) on port **3000**.

Let’s check to see if it’s working. Open [http://localhost:3000](http://localhost:3000/) from your browser.