**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.

**Create a Next.js App**

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

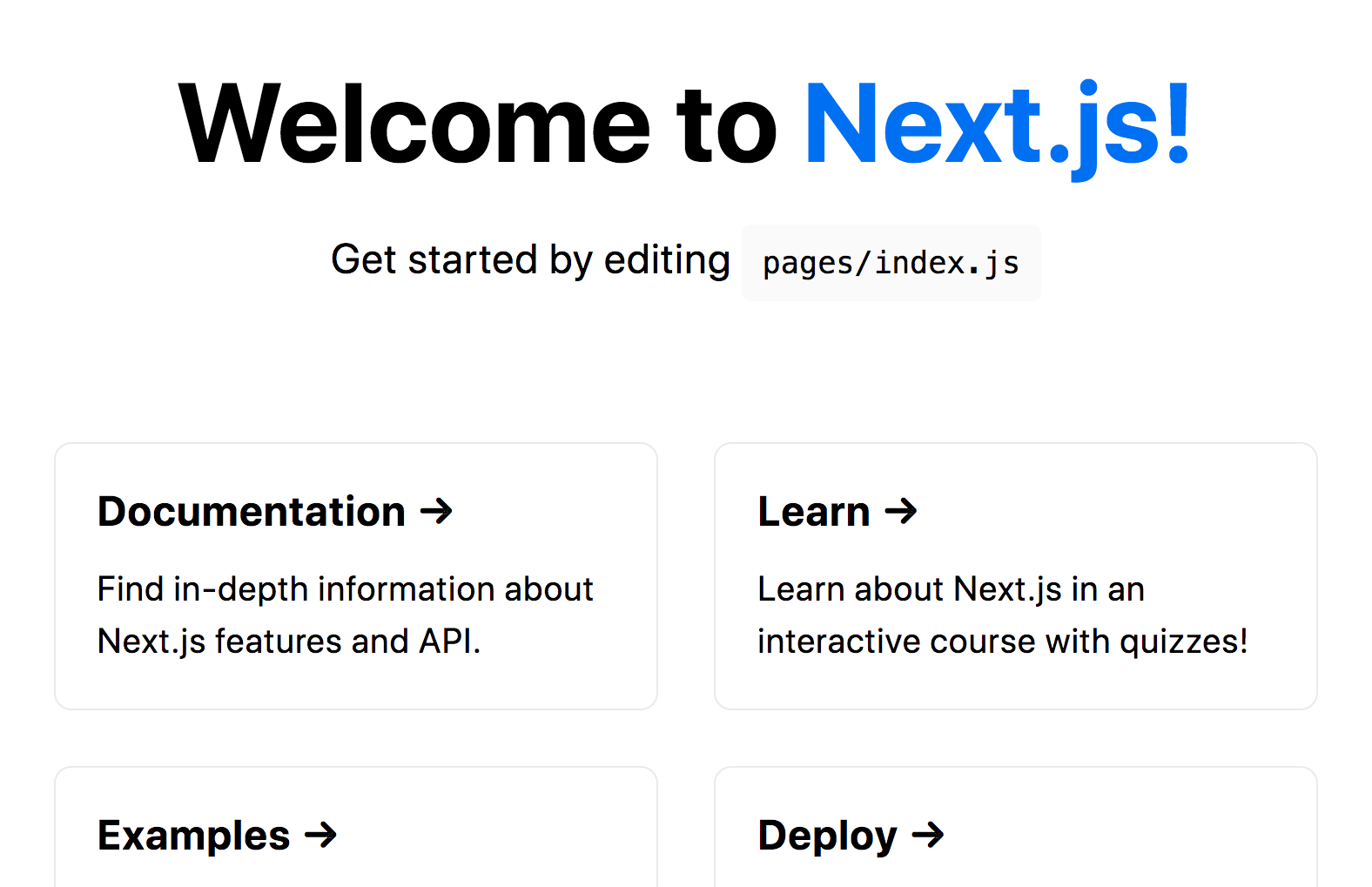
[2](https://nextjs.org/learn/basics/create-nextjs-app/setup)

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

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

**Welcome to Next.js**

You should see a page like this when you access [http://localhost:3000](http://localhost:3000/). This is the starter template page which shows some helpful information about Next.js.



**Help is available:** If you get stuck, you can reach out to the community on [GitHub Discussions](https://github.com/vercel/next.js/discussions).

Let’s try to edit this page next!

**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)

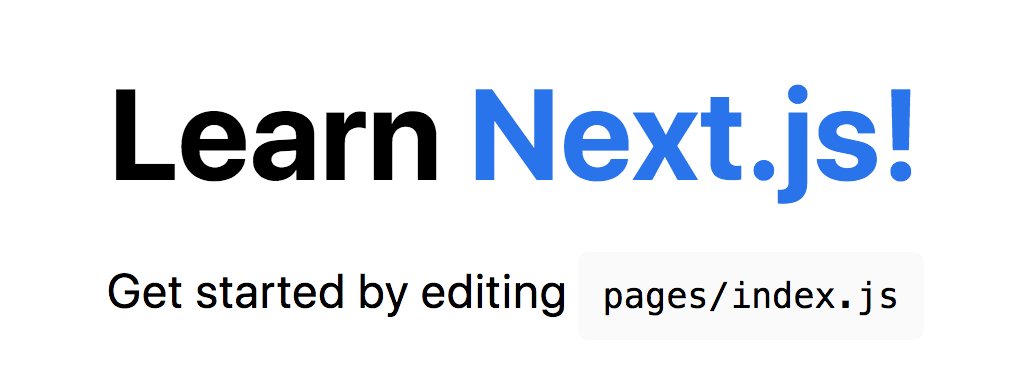
[**4**](https://nextjs.org/learn/basics/create-nextjs-app/editing-the-page)

**Editing the Page**

Let’s try editing the starter page.

* Make sure the Next.js development server is still running.
* Open pages/index.js with your text editor.
* Find the text that says **“Welcome to”** under the <h1> tag and change it to **“Learn”**.
* Save the file.

As soon as you save the file, the browser automatically updates the page with the new text:



The Next.js development server has [Fast Refresh](https://nextjs.org/docs/basic-features/fast-refresh) enabled. When you make changes to files, Next.js automatically applies the changes in the browser almost instantly. No refresh needed! This will help you iterate on your app quickly.

**Next Up: Creating Pages**

Great job! That’s it for the first lesson.

In the next lesson, we’ll talk about *creating more pages and navigating between pages*.

You should keep the development server running, but if you want to restart it, hit Ctrl + c to stop the server.

**Navigate Between Pages**

[**1**](https://nextjs.org/learn/basics/navigate-between-pages)

[2](https://nextjs.org/learn/basics/navigate-between-pages/setup)

[3](https://nextjs.org/learn/basics/navigate-between-pages/pages-in-nextjs)

[4](https://nextjs.org/learn/basics/navigate-between-pages/link-component)

[5](https://nextjs.org/learn/basics/navigate-between-pages/client-side)

So far, the Next.js app we created only has one page. Websites and web applications generally have many different pages.

Let's explore how to add more pages to our application.

**What You’ll Learn in This Lesson**

In this lesson, you will:

* Create a new page using the integrated [file system routing](https://nextjs.org/docs/routing/introduction).
* Learn how to use the [Link](https://nextjs.org/docs/api-reference/next/link) component to enable client-side navigation between pages.
* Learn about built-in support for code splitting and prefetching.

If you’re looking for detailed documentation on Next.js routing, take a look at the [routing documentation](https://nextjs.org/docs/routing/introduction).

**Navigate Between Pages**

[1](https://nextjs.org/learn/basics/navigate-between-pages)

[**2**](https://nextjs.org/learn/basics/navigate-between-pages/setup)

[3](https://nextjs.org/learn/basics/navigate-between-pages/pages-in-nextjs)

[4](https://nextjs.org/learn/basics/navigate-between-pages/link-component)

[5](https://nextjs.org/learn/basics/navigate-between-pages/client-side)

**If you’re continuing from the previous lesson,** you can skip this page. Click the button below to go to the next page.

[Next](https://nextjs.org/learn/basics/navigate-between-pages/pages-in-nextjs)

**Download Starter Code (Optional)**

If you’re NOT continuing from the previous lesson, you can download, install, and run the starter code for this lesson below. This sets up a nextjs-blog directory such that it’s identical to the result of the previous lesson.

Again, this is NOT necessary if you’ve just finished the previous lesson.

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

Then follow the instructions from the command output (cd into the directory and start the development server).

**Navigate Between Pages**

[1](https://nextjs.org/learn/basics/navigate-between-pages)

[2](https://nextjs.org/learn/basics/navigate-between-pages/setup)

[**3**](https://nextjs.org/learn/basics/navigate-between-pages/pages-in-nextjs)

[4](https://nextjs.org/learn/basics/navigate-between-pages/link-component)

[5](https://nextjs.org/learn/basics/navigate-between-pages/client-side)

**Pages in Next.js**

In Next.js, a page is a React Component exported from a file in the [pages directory](https://nextjs.org/docs/basic-features/pages).

Pages are associated with a route based on their **file name**. For example, in development:

* pages/index.js is associated with the / route.
* pages/posts/first-post.js is associated with the /posts/first-post route.

We already have the pages/index.js file, so let’s create pages/posts/first-post.js to see how it works.

**Create a New Page**

Create the posts directory under pages.

Create a file called first-post.js inside the posts directory with the following content:

export default function FirstPost() {

return <h1>First Post</h1>;

}

The component can have any name, but you must export it as a default export.

Now, make sure that the development server is running and visit <http://localhost:3000/posts/first-post>. You should see the page:



This is how you can create different pages in Next.js.

Simply create a JS file under the [pages directory](https://nextjs.org/docs/basic-features/pages), and the path to the file becomes the URL path.

In a way, this is similar to building websites using HTML or PHP files. Instead of writing HTML you write JSX and use React Components.

Let's add a link to the newly added page so that we can navigate to it from the homepage.

**Navigate Between Pages**

[1](https://nextjs.org/learn/basics/navigate-between-pages)

[2](https://nextjs.org/learn/basics/navigate-between-pages/setup)

[3](https://nextjs.org/learn/basics/navigate-between-pages/pages-in-nextjs)

[**4**](https://nextjs.org/learn/basics/navigate-between-pages/link-component)

[5](https://nextjs.org/learn/basics/navigate-between-pages/client-side)

**Link Component**

When linking between pages on websites, you use the <a> HTML tag.

In Next.js, you can use the Link Component [next/link](https://nextjs.org/docs/api-reference/next/link) to link between pages in your application. <Link> allows you to do client-side navigation and accepts [props](https://nextjs.org/docs/api-reference/next/link#:~:text=Link%20accepts%20the%20following%20props%3A) that give you better control over the navigation behavior.

**Using <Link>**

First, open pages/index.js, and import the Link component from [next/link](https://nextjs.org/docs/api-reference/next/link) by adding this line at the top:

import Link from 'next/link';

Then find the h1 tag that looks like this:

<h1 className="title">

Welcome to <a href="https://nextjs.org">Next.js!</a>

</h1>

And change it to:

<h1 className="title">

Read <Link href="/posts/first-post">this page!</Link>

</h1>

Next, open pages/posts/first-post.js and replace its content with the following:

import Link from 'next/link';

export default function FirstPost() {

return (

<>

<h1>First Post</h1>

<h2>

<Link href="/">Back to home</Link>

</h2>

</>

);

}

As you can see, the Link component is similar to using <a> tags, but instead of <a href="…">, you use <Link href="…">.

**Note:** Before Next.js 12.2, it was required that the Link component wrapped an <a> tag, but [this is not required in versions 12.2 and above](https://nextjs.org/blog/next-12-2#:~:text=next/link%20no%20longer%20requires%20manually%20adding%20%3Ca%3E%20as%20a%20child.%20You%20can%20now%20opt%20into%20this%20behavior%20in%20a%20backward%2Dcompatible%20way.).

Let’s check to see if it works. You should now have a link on each page, allowing you to go back and forth:



[Prev](https://nextjs.org/learn/basics/navigate-between-pages/pages-in-nextjs)[Next](https://nextjs.org/learn/basics/navigate-between-pages/client-side)