NORDIC
< CODER >

React ADVANCED

NEXT.JS

#Trương Thành Tài

#Frontend Technical Lead ekino.

> npx truthtai

WHAT IS NEXTJS ?

Next.js: The React Framework

- An intuitive page-based routing system (with support for dynamic routes)
- Pre-rendering, both static generation (SSG) and server-side rendering (SSR) are supported on a per-page basis
- Automatic code splitting for faster page loads
- Client-side routing with optimized prefetching
- Built-in CSS and Sass support, and support for any CSS-in-JS library
- Development environment which supports Hot Module Replacement
- API routes to build API endpoints with Serverless Functions
- Fully extendable

NORDIC
< CODER >

NEXT.JS

React ADVANCED

WHAT IS CREATE REACT APP ?

Create React App is an officially supported way to create single-page React applications.

ADVANCED

SSR vs. CSR vs SSG

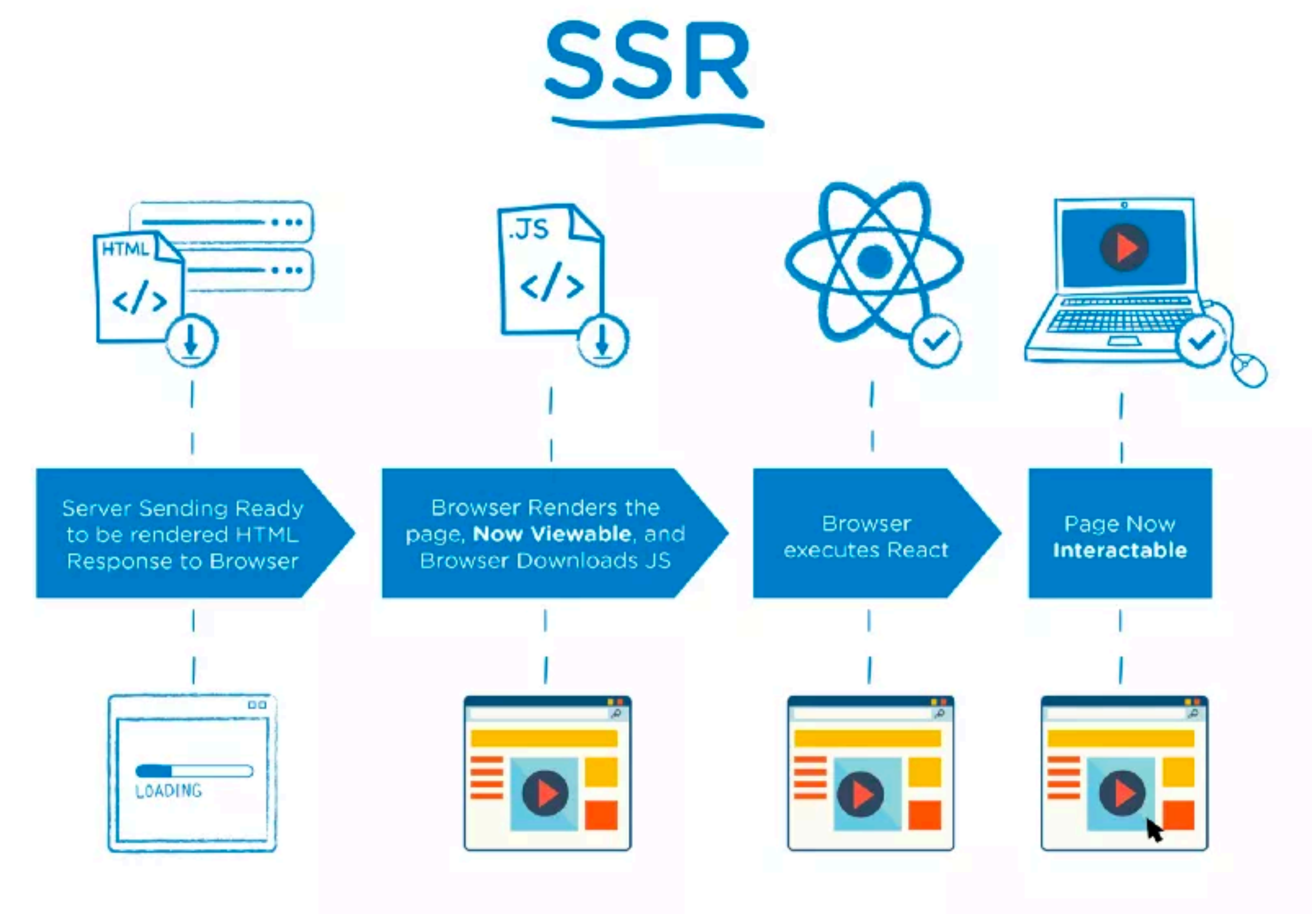
Next.js is one way that you can leverage React to support server-side rendering (Server-Side-Rendering - SSR).

Create React App is one way that you can leverage React to support client-side rendering (Client-side-rendering - CSR).

All the pages are generated at **build time** as static pages.
(Static Site Generation - SSG)

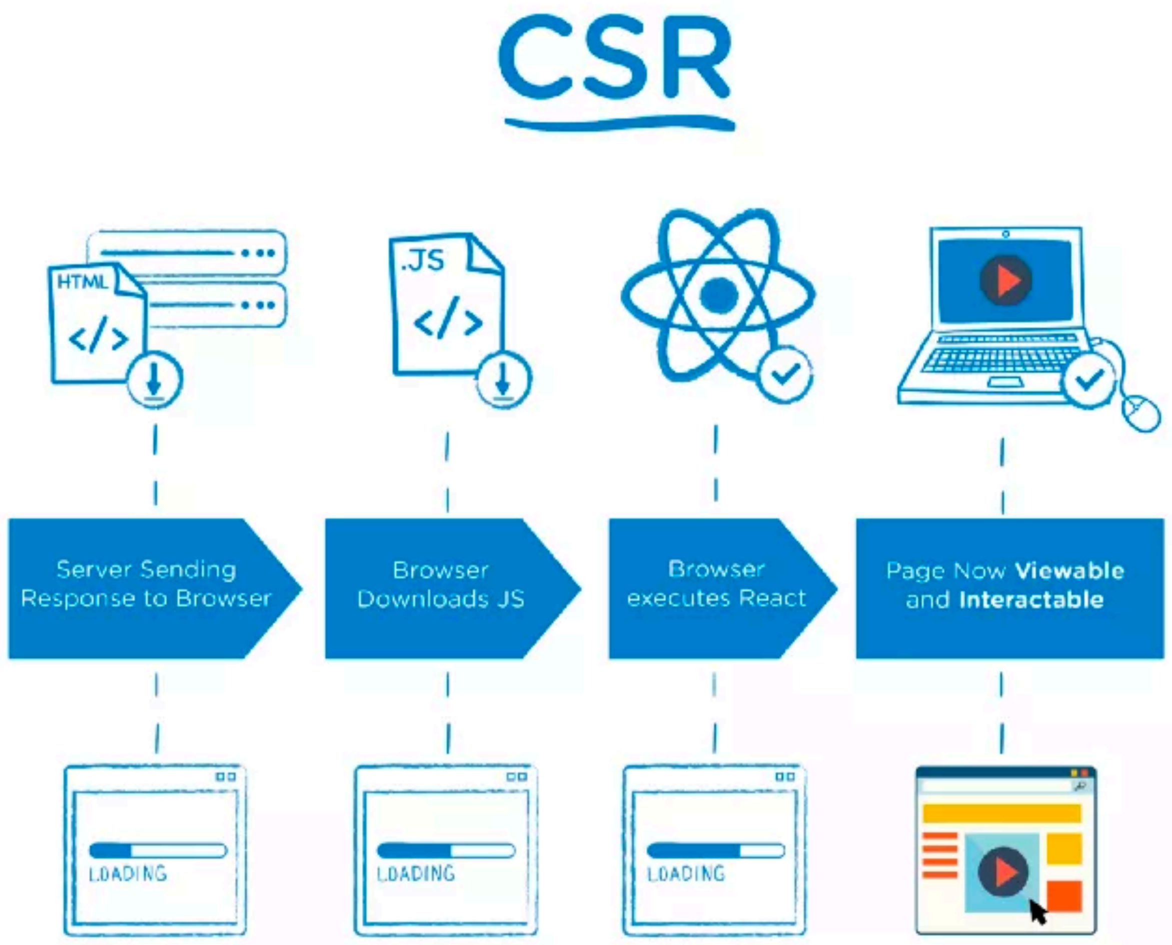
ADVANCED

SSR

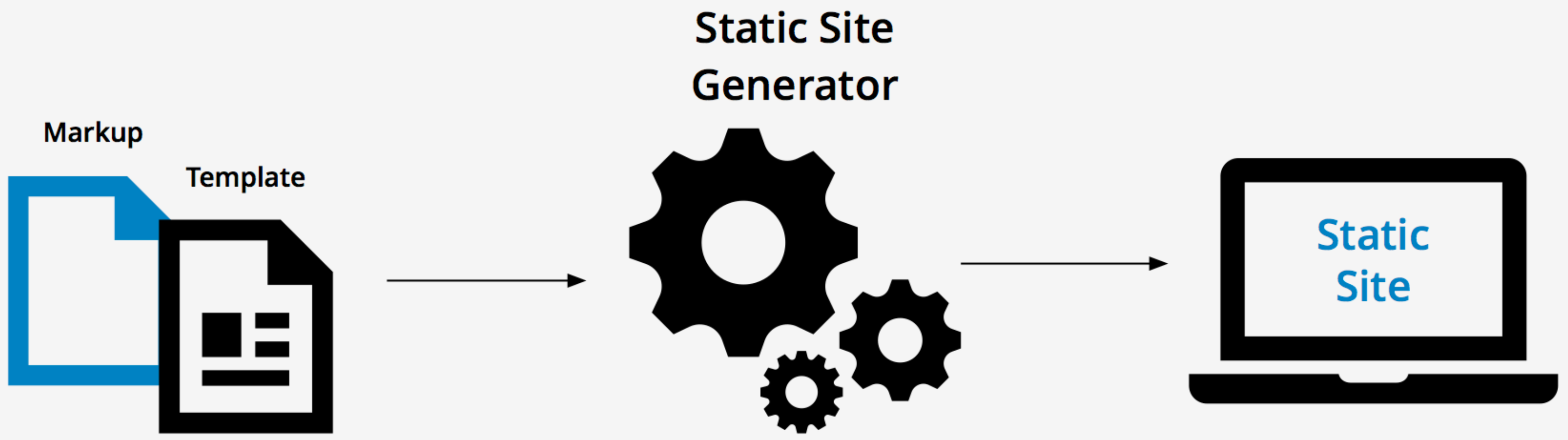

SSR
HTML
.JS
Server Sending Ready to be rendered HTML Response to Browser
Browser Renders the page, **Now Viewable**, and Browser Downloads JS
Browser executes React
Page Now **Interactable**
LOADING

ADVANCED

CSR



SSG



Static Site Generator

ADVANCED

NEXTJS – PAGE - ROUTER

In Next.js, a page is a React Component exported from a **.js, .jsx, .ts, or .tsx** file in the **pages** directory. Each page is associated with a route based on its file name.

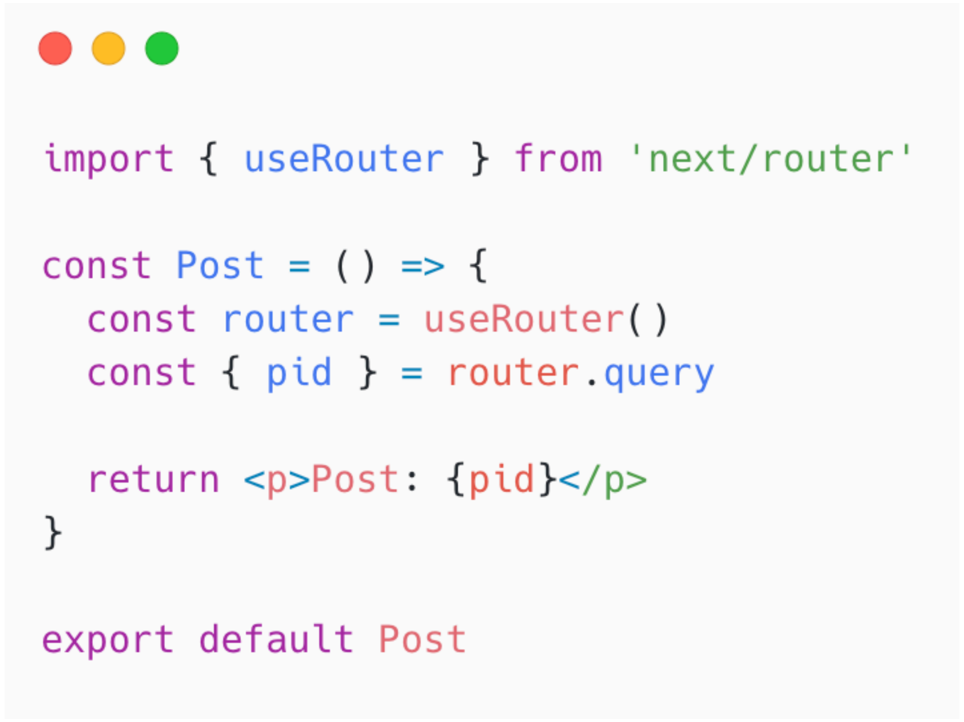
Example: If you create **pages/about.js** that exports a React component like below, it will be accessible at **/about**.

https://nextjs.org/docs/basic-features/pages

NORDIC <CODER> NEXT.JS React ADVANCED

NEXTJS – PAGE - DYNAMIC ROUTES

Consider the following page **pages/post/[pid].tsx**

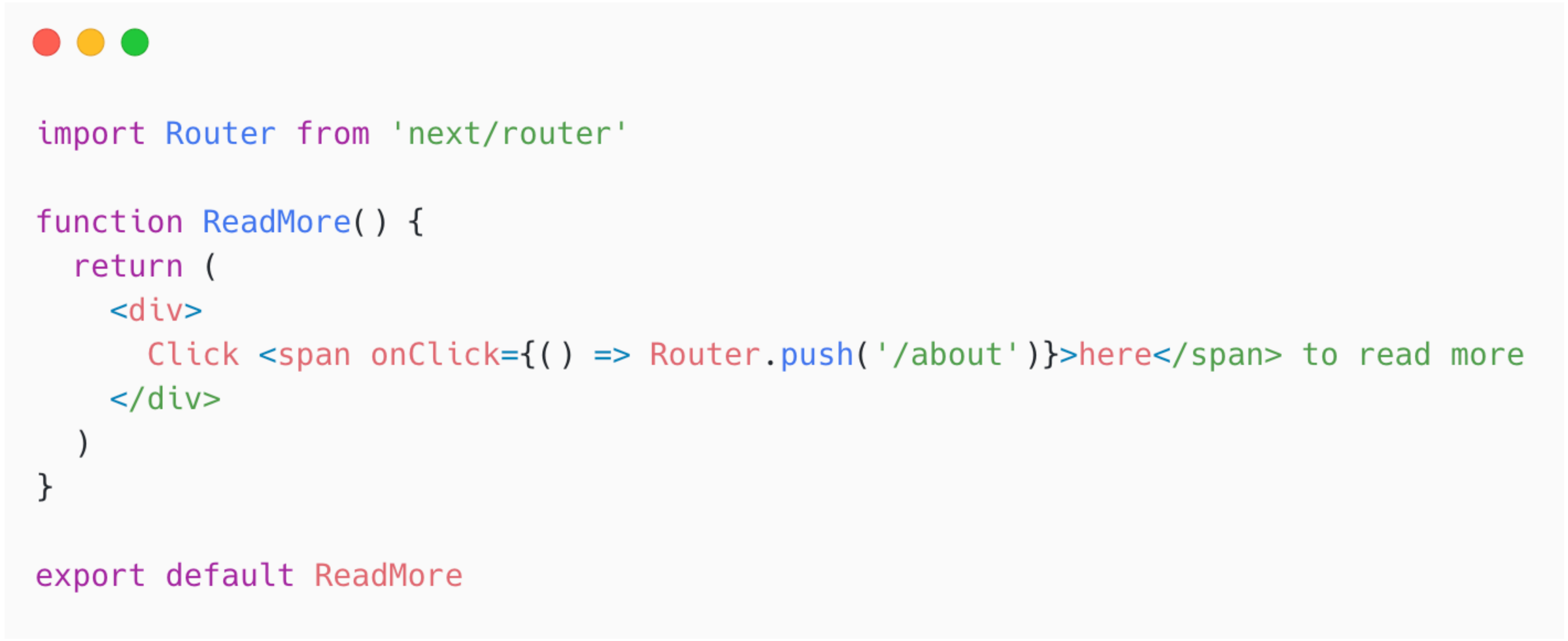


```
import { useRouter } from 'next/router'

const Post = () => {
  const router = useRouter()
  const { pid } = router.query

  return <p>Post: {pid}</p>
}

export default Post
```

https://nextjs.org/docs/routing/dynamic-routes

ADVANCED

NEXTJS – PAGE - IMPERATIVELY



```
import Router from 'next/router'

function ReadMore() {
  return (
    <div>
      Click <span onClick={() => Router.push('/about')}>here</span> to read more
    </div>
  )
}

export default ReadMore
```

https://nextjs.org/docs/routing/imperatively

NEXTJS – PRE RENDERING

1. **Static Generation** (Recommended): The HTML is generated at build time and will be reused on each request.
2. **Server-side Rendering**: The HTML is generated on each request.

https://nextjs.org/docs/basic-features/pages

ADVANCED

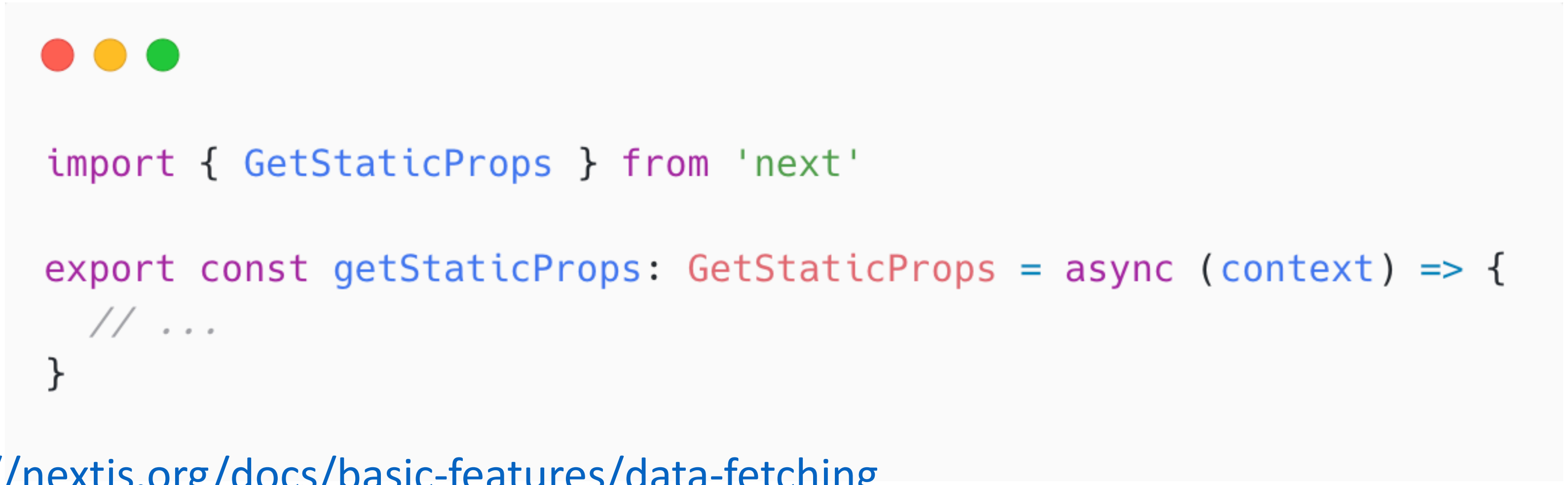
NEXTJS – FETCH DATA FOR PRE-RENDERING

1. **getStaticProps** (Static Generation): Fetch data at build time.
2. **getStaticPaths** (Static Generation): Specify dynamic routes to pre-render based on data.
3. **getServerSideProps** (Server-side Rendering): Fetch data on each request.

https://nextjs.org/docs/basic-features/data-fetching

NEXTJS – FETCH DATA FOR PRE-RENDERING / getstaticprops

1. The data required to render the page is available at build time ahead of a user’s request.
2. The data comes from headless CMS.
3. The data can be publicly cached (not user-specific).
4. The page must be pre-rendered (for SEO) and be very fast — getStaticProps generates HTML and JSON files, both of which can be cached by a CDN for performance.



```
import { GetStaticProps } from 'next'

export const getStaticProps: GetStaticProps = async (context) => {
  // ...
}
```

https://nextjs.org/docs/basic-features/data-fetching

NORDIC < CODER >

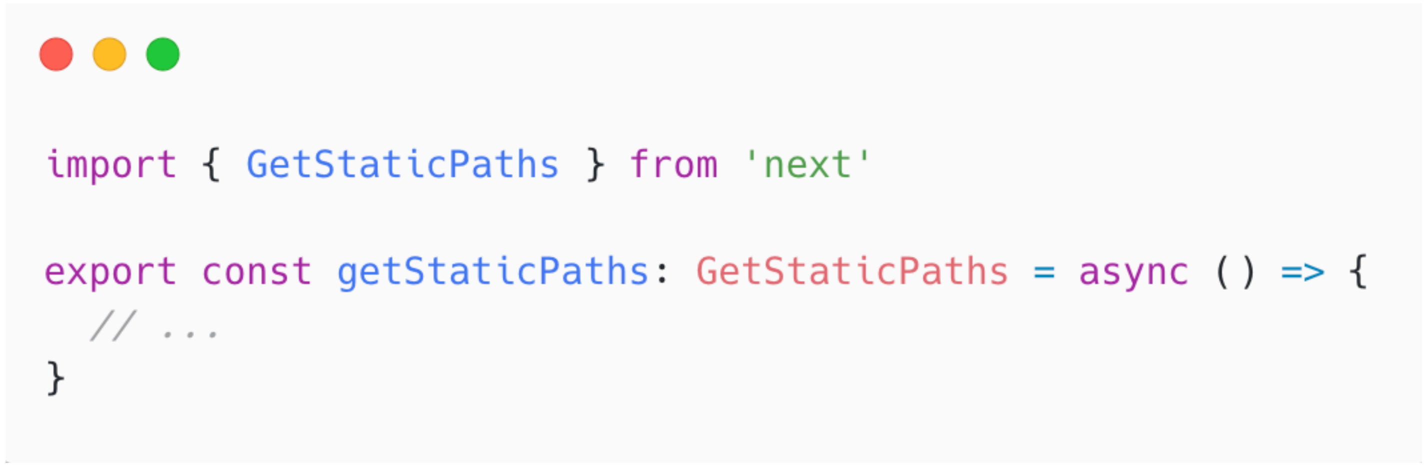
NEXT.JS

React

ADVANCED

NEXTJS – FETCH DATA FOR PRE-RENDERING / getStaticPaths

1. You should use **getStaticPaths** if you're statically pre-rendering pages that use dynamic routes.



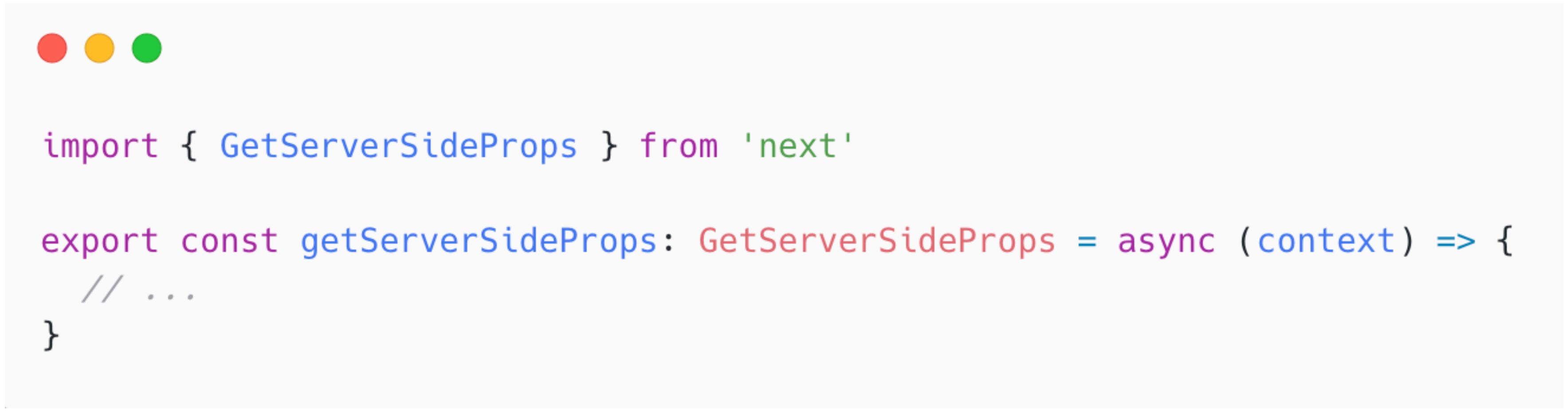
```
import { GetStaticPaths } from 'next'

export const getStaticPaths: GetStaticPaths = async () => {
  // ...
}
```

https://nextjs.org/docs/basic-features/data-fetching

ADVANCED

NEXTJS – FETCH DATA FOR PRE-RENDERING / getServerSideProps

1. You should use **getServerSideProps** only if you need to pre-render a page whose data must be fetched at request time. Time to first byte (TTFB) will be slower than **getStaticProps** because the server must compute the result on every request, and the result cannot be cached by a CDN without extra configuration.



```
import { GetServerSideProps } from 'next'

export const getServerSideProps: GetServerSideProps = async (context) => {
  // ...
}
```

https://nextjs.org/docs/basic-features/data-fetching

ADVANCED

LAB 1

Create a products page by using NextJS

ADVANCED

LAB 2 + HOMEWORK

1. Create product detail page
2. Click on specific product item, then we will navigate the page to product detail page by using NextJS Router
3. If we access to product detail page in browser (eg: abc.com/product/12345) directly, it should be loaded data by using server side.