Cheatsheets / **Intro to Next.js**

Next.js Core Concepts

Next.js

Next.js is a React framework that supports the development of robust web applications. It provides key features such as routing, rendering, data fetching, server components, and optimization. These features enable high-performing, scalable, and search-engine-friendly web applications.

Next.js Rendering

Rendering is the process of converting code into viewable user interfaces. In Next.js, different rendering strategies can be adopted depending on the application's requirements.

Next.js Rendering Strategies

Next.js enables developers to employ various rendering strategies, allowing the creation of hybrid web applications. These strategies allow developers to decide the rendering environment for individual components. The primary environments supported by Next.js are server-side and client-side rendering.

Server-side Rendering

In server-side rendering, the server infrastructure is used to render the webpage content. Then, a viewable, non-interactive webpage is sent to the client.
In Next.js, components are server-side rendered by default using server components. No additional configuration is necessary.

Client-side Rendering

In client-side rendering, the server sends the client all the necessary files for a webpage. The client uses the instructions to render the components on the browser and enable interactivity.
In Next.js, a component can be marked for client-side rendering with the `'use client'` directive.

```
'use client'
function MyComponent() {
  return <div>Hello, World!</div>;
}
```

`create-next-app`

A Next.js project can be created with the CLI tool `create-next-app`. It starts an interactive setup experience and generates a basic project structure and configuration when used.

Next.js App Router

Next.js uses the App Router, providing routing for the web application based on the `/app` directory. Each folder name determines a route that exists. To make the route accessible, a `page.tsx` file must live in the directory.
To make the `/` path accessible, the file structure must look like this:

```
├── app
|    ├── page.tsx
```

CSS Modules

Next.js supports multiple ways of styling the web app. It supports CSS modules, allowing a locally scoped styling method.

```
import styles from
'./MyComponent.module.css';
function MyComponent() {
  return <div className=
{styles.myClass}>Hello World</div>;
}
```

Next.js Hydration

After webpages are server-side rendered, they are made interactive on the client through hydration. Hydration executes the bundled JavaScript, attaching event handlers and linking the React components to their HTML counterparts. Reconciliation is performed to compare the results made on the client with those sent by the server, ensuring they’re in sync.

Print Share ▼