React server components

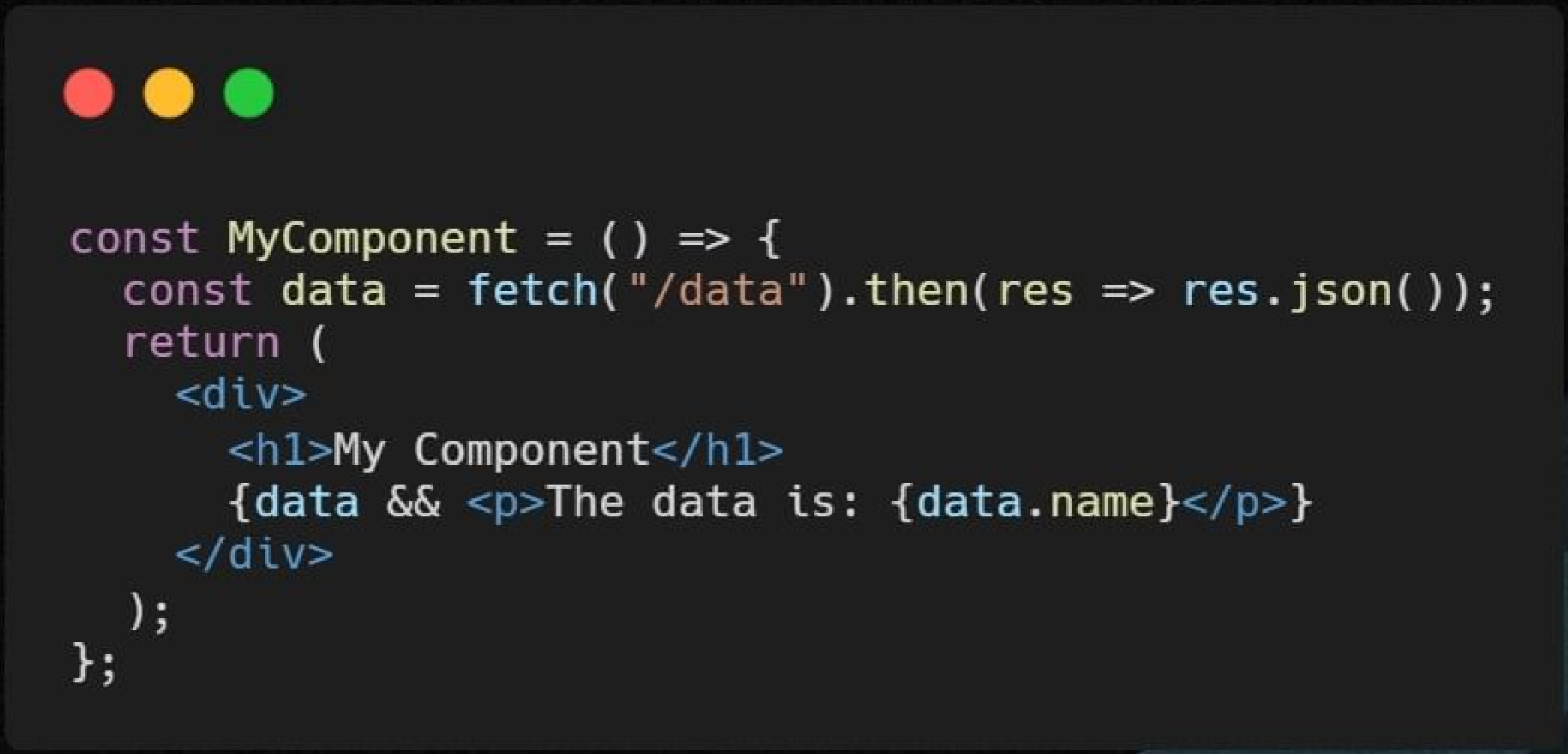
codewithsloba.com

Swipe ⟶

What are React Server Components?

React Server Components are a way to **render** React components **on the server** before sending them to the client.

This can improve performance and SEO, and it can also make your application more predictable.



```
const MyComponent = () => {
  const data = fetch("/data").then(res => res.json());
  return (
    <div>
      <h1>My Component</h1>
      {data && <p>The data is: {data.name}</p>}
    </div>
  );
};
```

Swipe ⟶

How do React Server Components work?

When you use React Server Components, the first time a user visits your page, the React components are rendered on the server.

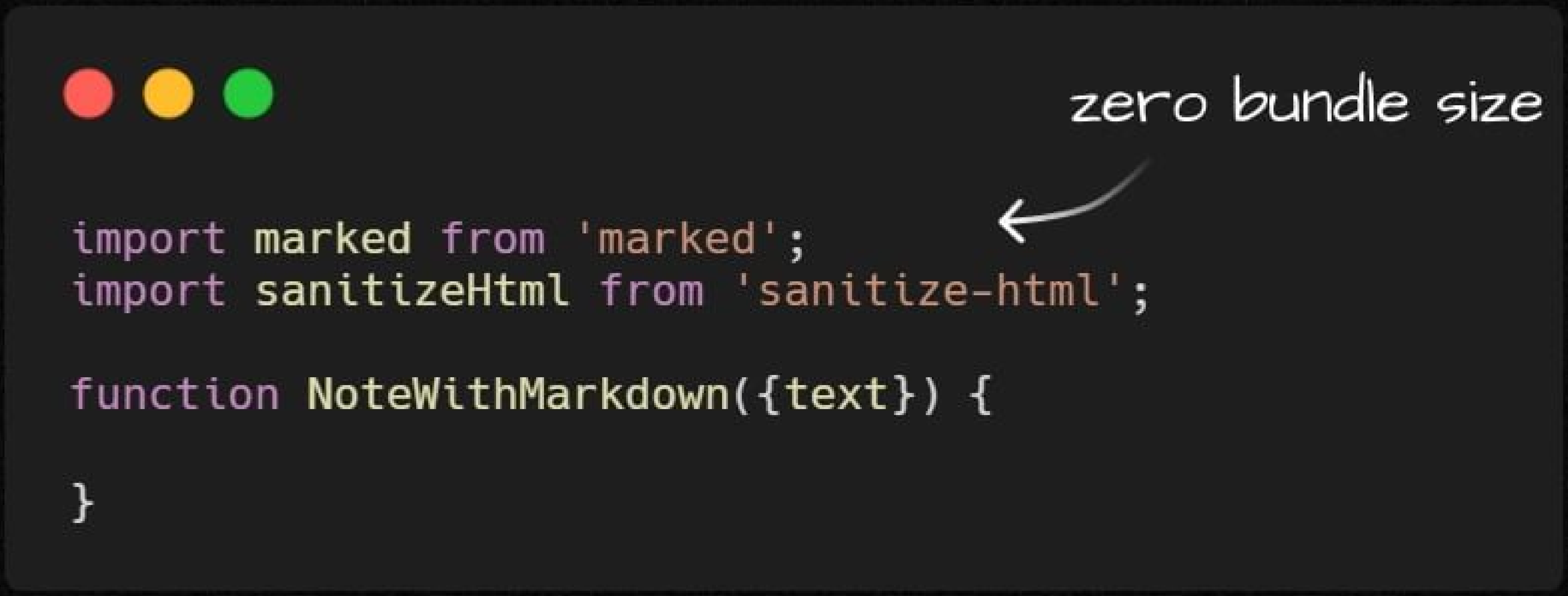
This means that the browser doesn't have to do any work to render the page, which can improve performance.

Additionally, the server can pre-fetch any data that the React components need, which can also improve performance.

Swipe ⟶

Benefits

You can freely use third-party packages in Server Components while incurring **zero impact on bundle size**.



zero bundle size

```
import marked from 'marked';
import sanitizeHtml from 'sanitize-html';

function NoteWithMarkdown({text}) {

}
```

swipe for more

With complete access to backend:

1. 🔒 Server components can help to improve the security of your application by preventing unauthorized access to your backend data.
2. Make queries directly on server and avoid client side load.
3. Access file system of your server. 📁

Avoid Waterfalls:

Move the logic part of the code to RCSs so there are no more fetch calls from client to server.

This reduces request latency and increases performance.

Balancing Server and Client Components:
Utilize RSCs for dynamic data fetching and Client Components for rich interactivity.

Automatic code splitting:
All client component imports are treated as possible code split points.

Limitations

Avoid using lifecycle hooks like useState, useReducer, useEffect as server components are rendered on the server.

Avoid using utility functions that depend on browser only API's.

codewithsloba.com

Get a weekly digest of my tips and tutorials by subscribing now.