Tamagui & NativeWind for React Native

## **Introduction**

Modern React Native apps aim for:

* Code reusability across platforms
* Performance
* Developer productivity

Two powerful libraries help accomplish this:

|  |  |
| --- | --- |
| **Library** | **Description** |
| **Tamagui** | UI kit and styling system focused on performance |
| **NativeWind** | Tailwind-style utility classes for React Native apps |

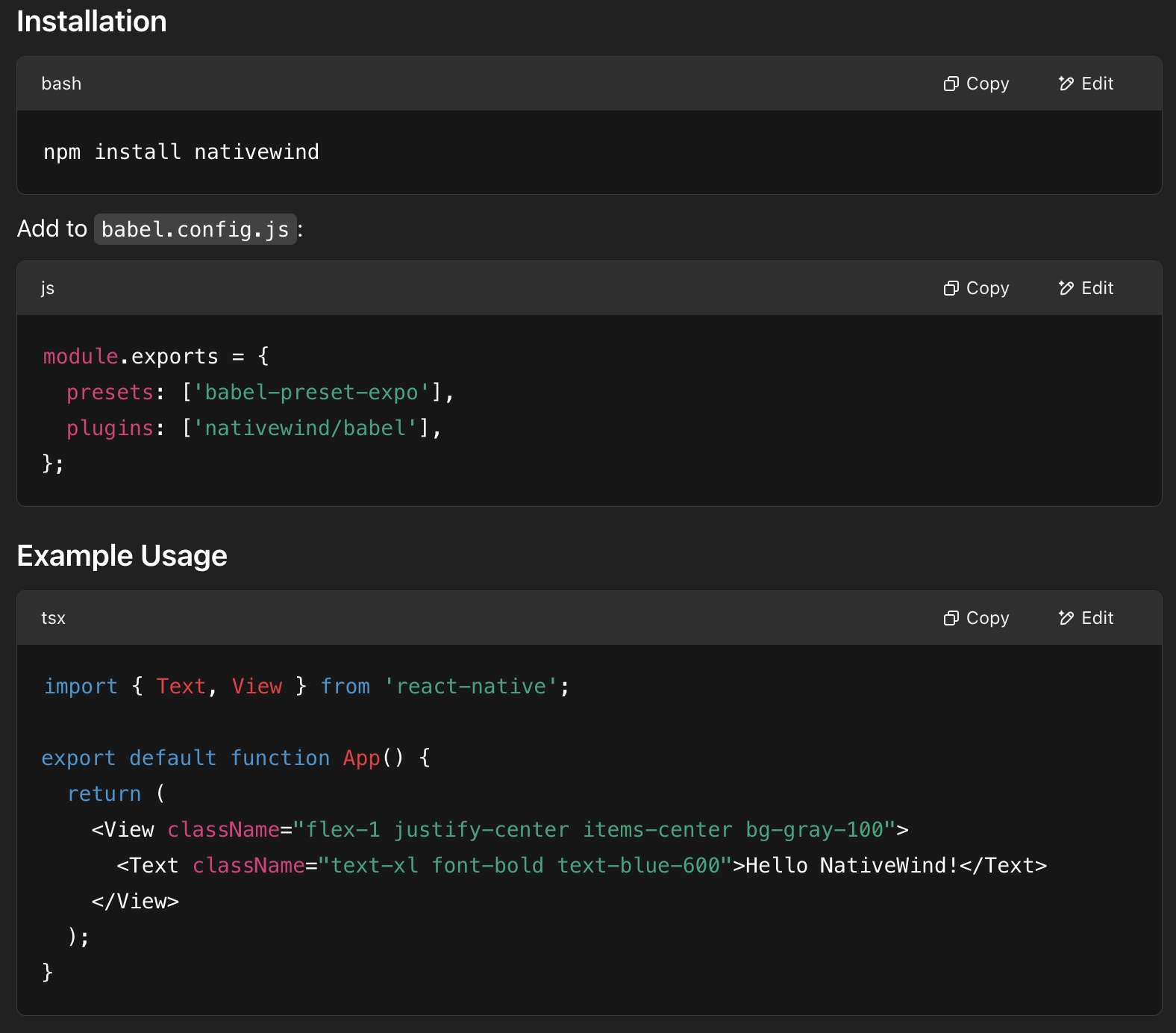
These tools enable **fast UI development**, with **cross-platform consistency** and **minimal runtime cost**.

## **1. What is NativeWind?**

**NativeWind** brings the **Tailwind CSS syntax** to React Native. Instead of writing complex StyleSheet.create() code, you use classes like in HTML/JSX.

### **Benefits:**

* Developer speed with utility-first syntax
* Responsive designs using media queries
* Works with **Expo**, **React Native CLI**, **Web** via react-native-web

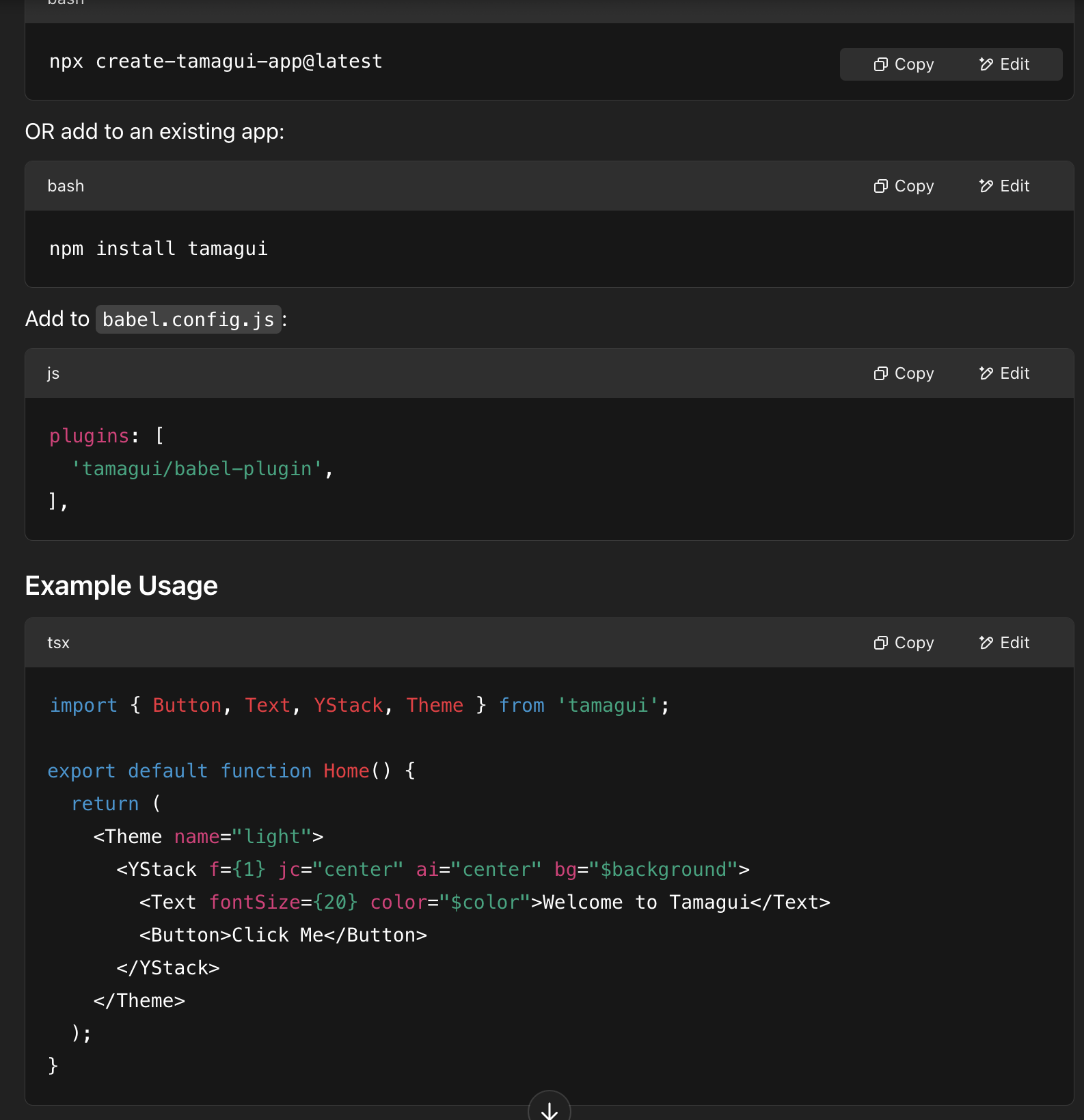


## **2. What is Tamagui?**

**Tamagui** is a UI kit and styling engine for **React Native** and **React Native Web**, built for **performance** and **design systems**.

### **Key Features:**

* Write styles like Tailwind, but with props
* Pre-compiled styles for runtime speed
* Theming support out of the box
* Comes with a set of UI components (like Button, Stack, Theme)



|  |  |  |
| --- | --- | --- |
| **Feature** | **Tamagui** | **NativeWind** |
| Syntax | Props-based (bg="$color") | Tailwind class-based (bg-blue-500) |
| Performance | Pre-compiled styles (very fast) | Runtime classes (faster than StyleSheet) |
| Component library | Yes | No (just styling) |
| Theming | Built-in | Requires manual setup |
| Web support | Yes | Yes |