Build React apps with Blueprint UI toolkit

Use the Blueprint library and you'll never have to struggle to create an attractive, easily accessible website again.

Building a React app from scratch can be time consuming and challenging, especially when it comes to styling components. This is where Blueprint UI Toolkit comes into play. This toolkit is a collection of reusable user interface components that help you create a consistent and visually appealing look and feel for any app developed using React. Here's how to use Blueprint UI Toolkit to create a simple React app.

What is Blueprint UI Toolkit?

Blueprint UI Toolkit is a React UI component library. It contains a set of pre-made components that are easy to use & customize. You can use these pre-designed components right out of the box or modify them to fit each need.

Blueprint UI Toolkit components include Buttons, Forms, Modals, Navigation, & Tables. Using them can save you time and effort because you don't need to design and build each component from scratch.

Instructions for using Blueprint UI Toolkit

To start using Blueprint UI Toolkit, you need to create a React app.

After setting up the project, you can install Blueprint UI Toolkit using any selected Node package installer. To install Blueprint UI Toolkit using npm, run the following command in the terminal:

```
npm install @blueprintjs/core
```

To use yarn instead, run the following command:

```
yarn add @blueprintjs/core
```

Once you've installed the Blueprint UI Toolkit, you can use select components in your React app.

Once you've installed the Blueprint UI Toolkit, you can use select components in your React app.

Using components in Blueprint UI Toolkit

Before using this component, include the CSS file from the Blueprint UI Toolkit:

```
@import "normalize.css"; @import "@blueprintjs/core/lib/css/blueprint.css"; @impo
```

Add the above code block to the CSS file that applies Blueprint UI styles to its element.

For example, to add a button to the application, use **the Button** from the Blueprint UI Toolkit:

```
import React from "react"; import { Button } from "@blueprintjs/core"; function A
```

This block of code adds a button to the application with **a Button** . The Button component takes properties like **intent** , **text** , **icon** , **small** , and large .

The intent property defines the nature of the button, reflected in the background color. In this example, the button serves the purpose of making the background color green. Blueprint UI provides several core intents including basic (blue), success (green), warning (orange), and danger (red).

You can specify the text that appears inside the button using the text property. You can also add icons to the button using the **icon** property . Next to the icon is **the rightIcon** - add the icon to the right side of the button.

Finally, the large and **small** boolean properties specify the size of the button. **The large** property makes the node larger, while **small** makes the node smaller.

The initial block of code will create a button like this:

You can also use the **AnchorButton** component to create buttons in your application. **The AnchorButton** component is a specialized version of the Button component that is explicitly designed to be used as a link.

This component accepts many of the same properties as the Button component, including **text** , **large** , **small** , **intent** , **icon . It also accepts href** and **target** attributes .

The href attribute specifies the URL the button links to, and the target specifies the target window or frame for the link:

```
import React from "react"; import { AnchorButton } from "@blueprintjs/core"; func
```

The code block above renders an **AnchorButton** component . The element's href attribute value is 'https://example.com/' and the target value is '_blank', which means the link will open in a different browser tab or window.

Another essential component of Blueprint UI Toolkit is **the Card** . This is a reusable component that displays information in a visually appealing way.

The Card component has two properties **interactive** & **elevation** . The elevation property controls the shadow depth of the tag, with higher values ??creating a more prominent shadow effect.

The interactive property accepts a boolean value. When set to true, it allows hover and click interactions on the tag, allowing the tag to respond to user input.

For example:

```
import React from "react"; import { Card, Elevation } from "@blueprintjs/core";
```

This is a Card

This is some content in my card

```
 ); } export default App;
```

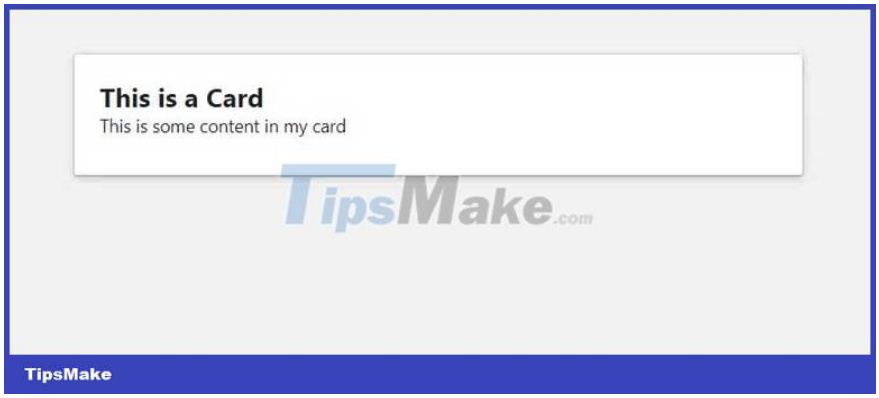
In this example, the **Card** component has a title and some content. interactive is set to true.

You also import the Elevation component from **@blueprintjs/core** . **Elevation** is an enum that defines a set of predefined values ??that you can use to set the element's shadow depth.

The following are the available values ??of the Elevation menu:

1. **Elevation.ZERO** : This value sets the shadow depth to 0, indicating the component doesn't have any shadows applied
2. **Elevation.ONE** : This value sets the depth of the shadow to 1.
3. **Elevation.TWO** : This value sets the depth of the shadow to 2.
4. **Elevation.THREE** : This value sets the depth of the shadow to 3.
5. **Elevation.FOUR** : This value sets the depth of the shadow to 4.
6. **Elevation.FIVE** : This value sets the depth of the shadow to 5.

Rendering the code block above will display an image on your screen that looks like this:



Blueprint UI also provides many other components, such as Alert, Popover, toast, etc. However, with the information provided, you can build a simple React application using Blueprint UI.

You finished reading the article "**Build React apps with Blueprint UI toolkit**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
