

Conclusion



Ashutosh Singh

- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top](#) ^



Installing and Using Chakra UI with React

Ashutosh Singh

Oct 30, 2020 • 9 Min read • 1,286 Views

Oct 30, 2020 • 9 Min read • 1,286 Views

Web Development

Front End Web Dev...

Client-side Framew...

React

Introduction

Compared to other React UI libraries like React Bootstrap, Material UI, etc., [Chakra UI](#) gives you much more power to style and customize components.

If you are a fan of Tailwind CSS, you will love Chakra UI since it follows the same minimalistic and modular approach as Tailwind CSS.

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top](#) ^

In this guide, we will discuss how to install, import, and use components from Chakra UI. We will use Chakra UI version 0.8 since version 1.0 is yet to be released [officially](#).



Installation

To install `chakra-ui`, run the following command in your project's root directory.

```
1 npm install @chakra-ui/core @emotion/core @emotion/styled emotion-theming
```

bash

Chakra UI uses [Emotion](#) for handling component styles. As you can see, you have to install peer dependencies yourself as they do not come with `chakra-ui` by default.

Next, wrap your React app in a `ThemeProvider`. Modify your `index.js` like this.

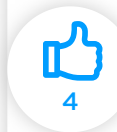
```
1 import React from "react";
2 import ReactDOM from "react-dom";
3 import App from "./App";
4 import { ThemeProvider } from "@chakra-ui/core";
5
```

JSX

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)



```
6   ReactDOM.render(  
7     <ThemeProvider>  
8       <App />  
9     </ThemeProvider>,  
10    document.getElementById("root")  
11  );
```

Importing Box Component

In this section, we will import the `Box` component from `chakra-ui`.

One of the coolest features of `chakra-ui` is that you can use shorthand for props. For example, instead of `width`, you can use `w`, which will work the same.

First, import `Box` from `chakra-ui` in `App.js` and use it inside the `App()` function.

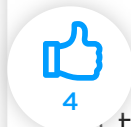
JSX

```
1   import React from "react";  
2   import { Box } from "@chakra-ui/core";  
3  
4   export default function App() {  
5     return (  
6       <div>  
7         <Box w="400px" h="400px">  
8           Hello World!  
9         </Box>
```

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)



In the above code, `w` and `h` are shorthand for `width` and `height` respectively.

Here is how this will look.

Hello World!

Styling Box Component

The `Box` component in the above example doesn't look like a box, just some text in some corner of the app. Add some background color using `bg` shorthand prop.

JSX

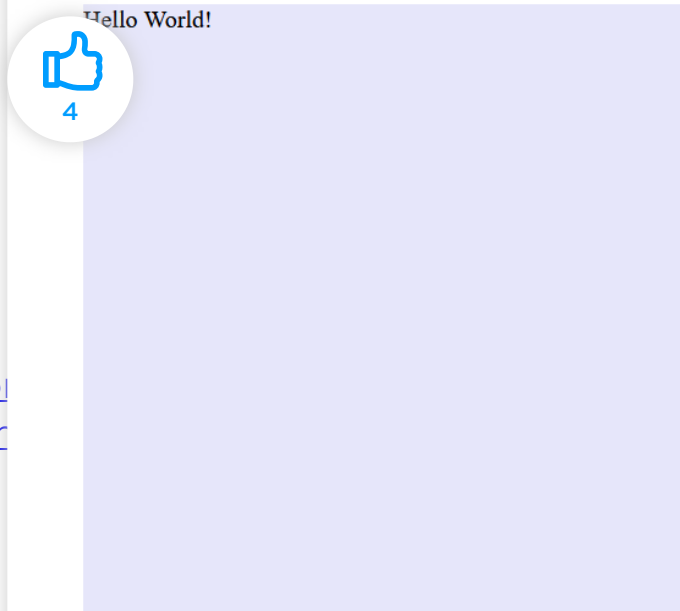
```
1    <Box bg="lavender" w="400px" h="400px">
2      Hello World!
3    </Box>
```

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)

Here is how this will look.



You can now differentiate the 400x400 box from the background.
Next, style the text and horizontally align it in the center of the box.

JSX

```
1    <Box
2      bg="lavender"
3      color="#2d383c"
4      fontSize="2rem"
5      textAlign="center"
6      fontFamily="Consolas"
7      w="400px"
8      h="400px"
9    >
10     Hello World!
11  </Box>
```

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)

You passed `color`, `textAlign`, `fontSize` and `fontFamily` prop to style the text inside the box. Here `1rem` is equal to `16px`.



Here is how this will look.



Now, add a border to the box using the `border` prop.

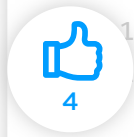
```
1    <Box
2      border="1px"
3      rounded="10px"
4      borderColor="gray.300"
5      boxShadow="md"
6      bg="lavender"
7      color="#2d383c"
8      fontSize="2rem"
9      textAlign="center"
10     fontFamily="Consolas"
```

JSX

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)



```
11     w="400px"  
12     h="400px"  
13     >  
14     Hello World!  
15   </Box>
```

Here `rounded` is shorthand for `border-radius`, and `boxShadow` is shorthand for `box-shadow`. You have used `md` as the value of `boxShadow` prop, which is equivalent to `16px`.

Here is how this will look.



To add some `margin` and `padding`, use the shorthand prop `p` and `m`. You can also use `px` and `py` where `px` sets the padding for

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)

`left` and `right`. Similarly, `py` sets it for `top` and `bottom`.

Also, change the text `Hello World!` to `Lorem Ipsum`.



JSX

```
1    <Box
2      m="8px"
3      p="8px"
4      border="1px"
5      rounded="10px"
6      borderColor="gray.300"
7      boxShadow="md"
8      bg="lavender"
9      color="#2d383c"
10     fontSize="2rem"
11     textAlign="center"
12     fontFamily="Consolas"
13     w="400px"
14     h="400px"
15   >
16     Lorem ipsum dolor sit amet, consectetur adipisicing elit. Molestias aper
17     doloremque exercitationem saepe, sed modi odit officia illum iste vel?
18     dignissimos pariatur, odit impedit iste aperiam facere atque iure!{" "}
19   </Box>
```

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)



Lorem ipsum dolor sit
amet, consectetur
adipiscing elit.
Molestias aperi-
am
doloremque
exercitationem saepe,
sed modi odit officia
illum iste vel? Rerum
dignissimos pariatur,
odit impedit iste
aperiam facere atque
iure!

As you can see, the text is overflowing the `Box` component; you can fix this by passing `overflow="hidden"` prop to `Box`.

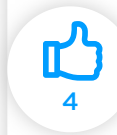
JSX

```
1    <Box  
2      m="8px"  
3      p="8px"  
4      border="1px"  
5      rounded="10px"  
6      borderColor="gray.300"  
7      boxShadow="md"  
8      bg="lavender"  
9      color="#2d383c"  
10     fontSize="2rem"  
11     textAlign="center"  
12     fontFamily="Consolas"
```

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)



```
13     w="400px"  
14     h="400px"  
15     overflow="hidden"  
16   >  
17     Lorem ipsum...  
18   </Box>
```

Here is how this will look.

Lorem ipsum dolor sit
amet, consectetur
adipiscing elit.
Molestias aperi-
am
doloremque
exercitationem saepe,
sed modi odit officia
illum iste vel? Rerum
dignissimos pariatur,
odit impedit iste
aperiam facere atque

`isTruncated` is another cool prop you can pass to truncate long texts.

```
1   <Box  
2     m="8px"  
3     p="8px"
```

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)



```
4      border="1px"
5      rounded="10px"
6      borderColor="gray.300"
7      boxShadow="md"
8      bg="lavender"
9      color="#2d383c"
10     fontSize="2rem"
11     textAlign="center"
12     fontFamily="Consolas"
13     w="400px"
14     h="400px"
15     isTruncated
16   >
17     Lorem ipsum ....
18   </Box>
```

Here is how this will look.

Lorem ipsum dolor sit...

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)



Adding Image to Box

In this section, you will add an image to the `Box` component in the above example. For this, you will use the `Image` and `Text` components.

Import `Image` and `Text` in your `App.js`.

```
1      import { Box, Image, Text } from "@chakra-ui/core";
```

JSX

Now add these components inside the `Box` component.

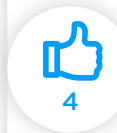
```
1      <Box
2        m="8px"
3        p="8px"
4        border="1px"
5        rounded="10px"
6        borderColor="gray.300"
7        boxShadow="md"
8        bg="lavender"
9        color="#2d383c"
10       fontSize="2rem"
11       textAlign="center"
12       fontFamily="Consolas"
13       w="400px"
```

JSX

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)



```
14     h="400px"  
15   >  
16   <Image  
17     rounded="0.5rem"  
18     src="https://finalspaceapi.com/img/gary_goodspeed.webp"  
19     alt="Gary Goodspeed"  
20   />  
21   <Text>Gary Goodspeed</Text>  
22 </Box>
```

Here is how this will look.



What if there are multiple boxes one after another?

Copy and paste this box multiple times.

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)



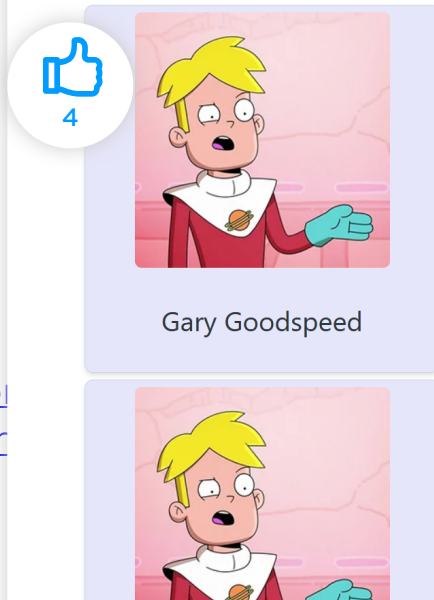
```
<div>
  <Box>
1    <Image
2      rounded="0.5rem"
3      src="https://finalspaceapi.com/img/gary_goodspeed.webp"
4      alt="Gary Goodspeed"
5    />
6    <Text>Gary Goodspeed</Text>
7  </Box>
8
9  <Box>
10   <Image
11     rounded="0.5rem"
12     src="https://finalspaceapi.com/img/gary_goodspeed.webp"
13     alt="Gary Goodspeed"
14   />
15   <Text>Gary Goodspeed</Text>
16 </Box>
17
18 <Box>
19   <Image
20     rounded="0.5rem"
21     src="https://finalspaceapi.com/img/gary_goodspeed.webp"
22     alt="Gary Goodspeed"
23   />
24   <Text>Gary Goodspeed</Text>
25 </Box>
26 </div>
27
28
```

Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)

Here is how this will look.



You can make the layout of this card system responsive by using the `SimpleGrid` component. First, import the `SimpleGrid` component in your `App.js`.

```
1 import { Box, Image, Text, SimpleGrid } from "@chakra-ui/core";
```

JSX

Now wrap all the `Box` components inside this `SimpleGrid` component.

Pass the `minChildWidth` prop to this `SimpleGrid` component, which will adjust the arrangement of the boxes according to `minChildWidth`.

Here the boxes are of 400x400px dimension, so you can pass `410px` as `minChildWidth`.

Conclusion



JSX



```
1    <SimpleGrid minChildWidth="410px">
2      <Box> ...</Box>
3      <Box> ...</Box>
4      <Box> ...</Box>
5      <Box> ...</Box>
6      <Box> ...</Box>
7      <Box> ...</Box>
8      <Box> ...</Box>
9      <Box> ...</Box>
10     <Box> ...</Box>
11  </SimpleGrid>
```

- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)

Here is how this will look.



Conclusion



- [Introduction](#)
- [Installation](#)
- [Importing Box Component](#)
- [Styling Box Component](#)
- [Adding Image to Box](#)
- [Conclusion](#)
- [Top ^](#)

There is still so much more that you can do to customize and style components in Chakra UI.



Conclusion

In this guide, we discussed how to install `chakra-ui` in a React app. We also covered how to import and customize different components from `chakra-ui`.

Chakra UI is the perfect choice for your React app when you want to customize components and want your app to have a unique look, which is difficult to achieve in other React UI libraries like React Bootstrap, Material UI, etc.

Here are some additional resources that can be helpful.

- [Chakra UI Docs](#)
- [Chakra UI GitHub](#)

Happy coding!