Version: 0.69

# State

There are two types of data that control a component: `props` and `state`. `props` are set by the parent and they are fixed throughout the lifetime of a component. For data that is going to change, we have to use `state`.

In general, you should initialize `state` in the constructor, and then call `setState` when you want to change it.

For example, let's say we want to make text that blinks all the time. The text itself gets set once when the blinking component gets created, so the text itself is a `prop`. The "whether the text is currently on or off" changes over time, so that should be kept in `state`.

State

Expo

```
import React, { useState, useEffect } from 'react';
import { Text, View } from 'react-native';

const Blink = (props) => {
  const [isShowingText, setIsShowingText] =
useState(true);

   useEffect(() => {
     const toggle = setInterval(() => {
       setIsShowingText(!isShowingText);
     }, 5000);

     return () => clearInterval(toggle);
  })

  if (!isShowingText) {
    return null;
  }

  return <Text>{props.text}</Text>;
}

const BlinkApp = () => {
  return (
    <View style={{marginTop: 50}}>
      <Blink text='I love to blink' />
```

Preview | My Device | iOS | Android | Web

In a real application, you probably won't be setting state with a timer. You might set state when you have new data from the server, or from user input. You can also use a state container like Redux or MobX to control your data flow. In that case you would use Redux or MobX to modify your state rather than calling `setState` directly.

When setState is called, BlinkApp will re-render its Component. By calling setState within the Timer, the component will re-render every time the Timer ticks.

State works the same way as it does in React, so for more details on handling state, you can look at the React.Component API. At this point, you may have noticed that most of our examples use the default text color. To customize the text color, you will have to learn about Style.

Edit this page

*Last updated on **Jun 22, 2022***