

Style

With React Native, you style your application using JavaScript. All of the core components accept a prop named `style`. The style names and values usually match how CSS works on the web, except names are written using camel casing, e.g. `backgroundColor` rather than `background-color`.

The `style` prop can be a plain old JavaScript object. That's what we usually use for example code. You can also pass an array of styles - the last style in the array has precedence, so you can use this to inherit styles.

As a component grows in complexity, it is often cleaner to use `StyleSheet.create` to define several styles in one place. Here's an example:

Style

^ Expo

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

const LotsOfStyles = () => {
  return (
    <View style={styles.container}>
      <Text style={styles.red}>just red</Text>
      <Text style={styles.bigBlue}>just bigBlue</Text>
      <Text style={[styles.bigBlue, styles.red]}>bigBlue,
then red</Text>
      <Text style={[styles.red, styles.bigBlue]}>red, then
bigBlue</Text>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    marginTop: 50,
  },
  bigBlue: {
    color: 'blue',
    fontWeight: 'bold',
    fontSize: 30,
  },
  red: {
    color: 'red',
  },
});
```

Preview ☐ My Device ☐ iOS ☐ Android ☒ Web

One common pattern is to make your component accept a `style` prop which in turn is used to style subcomponents. You can use this to make styles "cascade" the way they do in CSS.


There are a lot more ways to customize the text style. Check out the [Text component reference](#) for a complete list.

Now you can make your text beautiful. The next step in becoming a style expert is to [learn how to control component size](#).

Known issues

- [react-native#29308](#): In some cases React Native does not match how CSS works on the web, for example the touch area never extends past the parent view bounds and on Android negative margin is not supported.

Is this page useful?  

 Edit this page

Last updated on **Jun 21, 2023**