

Transforms

Transforms are style properties that will help you modify the appearance and position of your components using 2D or 3D transformations. However, once you apply transforms, the layouts remain the same around the transformed component hence it might overlap with the nearby components. You can apply margin to the transformed component, the nearby components or padding to the container to prevent such overlaps.

Example

Transforms

^ Expo

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

const App = () => (
  <SafeAreaView style={styles.container}>
    <ScrollView contentContainerStyle={styles.scrollContentContainer}>
      <View style={styles.box}>
        <Text style={styles.text}>Original Object</Text>
      </View>

      <View
        style={[
          styles.box,
          {
            transform: [{scale: 2}],
          },
        ]>
        <Text style={styles.text}>Scale by 2</Text>
      </View>

      <View
        style={[
          styles.box,
          {
            transform: [{scaleX: 2}],
          },
        ]>
        <Text style={styles.text}>Scale by 2</Text>
      </View>
    </ScrollView>
  </SafeAreaView>
);
```

Preview

My Device

iOS

Android

Web

Reference

Methods

transform()

transform accepts an array of transformation objects or space-separated string values. Each object specifies the property that will be transformed as the key, and the value to use in the transformation. Objects should not be combined. Use a single key/value pair per object.

The rotate transformations require a string so that the transform may be expressed in degrees (deg) or radians (rad). For example:

```
transform([{'rotateX': '45deg'}, {'rotateZ': '0.785398rad'}]);
```

The same could also be achieved using a space-separated string:

```
transform('rotateX(45deg) rotateZ(0.785398rad)');
```

The skew transformations require a string so that the transform may be expressed in degrees (deg). For example:


```
transform([{'skewX': '45deg'}]);
```

TYPE	REQUIRED
array of objects: {matrix: number[]}, {perspective: number}, {rotate: string}, {rotateX: string}, {rotateY: string}, {rotateZ: string}, {scale: number}, {scaleX: number}, {scaleY: number}, {translateX: number}, {translateY: number}, {skewX: string}, {skewY: string} or string	No

decomposedMatrix, rotation, scaleX, scaleY, transformMatrix, translateX, translateY

Deprecated. Use the `transform` prop instead.

Is this page useful?  

 Edit this page

Last updated on **Aug 17, 2023**