Version: 3.x

interpolateColor

> INFO
>
> This page was ported from an old version of the documentation.
>
> As we're rewriting the documentation some of the pages might be a little outdated.

Maps input range to output colors using linear interpolation. It works just like `interpolate` function but the output is color string in `rgba(r, g, b, a)` notation.

Arguments

`value` **[Float]**

Value from within the input range that should be mapped to a value from the output range.

`input range` **[Float[]]**

An array of Floats that contains points that indicate the range of the input value. Values in the input range should be increasing.

`output range` **[(string | number)[]]**

An array of colors (strings like `'red'`, `'#ff0000'`, `'rgba(255, 0, 0, 0.5)'` etc.) that contains points that indicate the range of the output value. It should have at least the same number of points as the input range.

`color space` **[String]**

Can be either `'RGB'` or `'HSV'`.

`options` **[Object]**

Object containing color interpolation options. Allowed parameters are listed below:

Options	Default	Description
gamma	2.2	Gamma parameter used in gamma correction.
useCorrectedHSVInterpolation	true	See Options explanation section

Returns

`interpolateColor` returns the color after interpolation from within the output range in `rgba(r, g, b, a)` format.

Options explanation

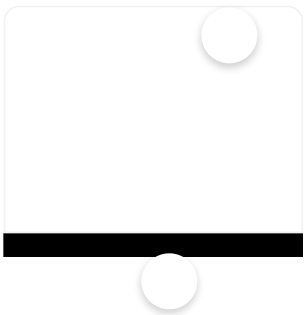
- *gamma* - Colors on web / mobile are expressed in sRGB colorspace which is gamma-corrected, that is non-linear. Operations on colors in non-linear space like addition will give wrong results. For example the interpolated color may appear darker than it should. That's why when interpolating we convert sRGB to linear space first and then convert the result back into non-linear sRGB space. Gamma correction is device-dependent but for most devices to convert from non-linear to linear space raising components to the power of gamma=2.2 is a good approximation. If you'd like to disable that you can always set gamma=1. A nice article on that if you'd like to know more: https://observablehq.com/@sebastien/srgb-rgb-gamma
- *useCorrectedHSVInterpolation* - Sometimes (for example when interpolating from yellow to purple) HSV interpolation goes through many other hues. This option allows to reduce the number of hues in such cases by treating HSV hues like a circular spectrum and choosing the shortest arc (so instead of going from yellow to purple through green and blue, it goes only through red).

Example

```
const Component = () => {
  const progress = useSharedValue(0);

  const animatedStyle = useAnimatedStyle(() => {
```

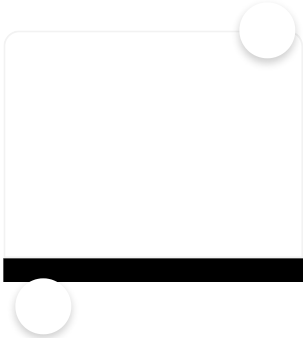
```
    return {
      backgroundColor: interpolateColor(
        progress.value,
        [0, 1],
        ['red', 'green']
      ),
    };
  });

  return (
    <View>
      <Animated.View style={[{ width: 100, height: 100 }, animatedStyle]} />
      <Button
        onPress={() => {
          progress.value = withTiming(1 - progress.value, { duration: 1000 });
        }}
        title="run animation"
      />
    </View>
  );
};
```

Playground



38ACDD

RUN ANIMATION



FFD61E

Show interpolation comparison

Colorspace RGB

Gamma 2.2

```
interpolateColor(
    sv.value,
    [0, 1],
    ['#38ACDD', '#FFD61E']
    'RGB',
    {
      gamma: 2.2,
    }
  )
```

Edit this page