import {Value} from './index';

/\*\*

\* Sass's [color type](https://sass-lang.com/documentation/values/colors).

\*

\* No matter what representation was originally used to create this color, all

\* of its channels are accessible.

\*

\* @category Custom Function

\*/

export class SassColor extends Value {

/\*\*

\* Creates an RGB color.

\*

\* @throws `Error` if `red`, `green`, and `blue` aren't between `0` and

\* `255`, or if `alpha` isn't between `0` and `1`.

\*/

constructor(options: {

red: number;

green: number;

blue: number;

alpha?: number;

});

/\*\*

\* Creates an HSL color.

\*

\* @throws `Error` if `saturation` or `lightness` aren't between `0` and

\* `100`, or if `alpha` isn't between `0` and `1`.

\*/

constructor(options: {

hue: number;

saturation: number;

lightness: number;

alpha?: number;

});

/\*\*

\* Creates an HWB color.

\*

\* @throws `Error` if `whiteness` or `blackness` aren't between `0` and `100`,

\* or if `alpha` isn't between `0` and `1`.

\*/

constructor(options: {

hue: number;

whiteness: number;

blackness: number;

alpha?: number;

});

/\*\* This color's red channel, between `0` and `255`. \*/

get red(): number;

/\*\* This color's green channel, between `0` and `255`. \*/

get green(): number;

/\*\* This color's blue channel, between `0` and `255`. \*/

get blue(): number;

/\*\* This color's hue, between `0` and `360`. \*/

get hue(): number;

/\*\* This color's saturation, between `0` and `100`. \*/

get saturation(): number;

/\*\* This color's lightness, between `0` and `100`. \*/

get lightness(): number;

/\*\* This color's whiteness, between `0` and `100`. \*/

get whiteness(): number;

/\*\* This color's blackness, between `0` and `100`. \*/

get blackness(): number;

/\*\* This color's alpha channel, between `0` and `1`. \*/

get alpha(): number;

/\*\*

\* Changes one or more of this color's RGB channels and returns the result.

\*/

change(options: {

red?: number;

green?: number;

blue?: number;

alpha?: number;

}): SassColor;

/\*\*

\* Changes one or more of this color's HSL channels and returns the result.

\*/

change(options: {

hue?: number;

saturation?: number;

lightness?: number;

alpha?: number;

}): SassColor;

/\*\*

\* Changes one or more of this color's HWB channels and returns the result.

\*/

change(options: {

hue?: number;

whiteness?: number;

blackness?: number;

alpha?: number;

}): SassColor;

}