

Background

Contents

Background color	1
Background gradient	1
Opacity	2
How it works	2
Example	2
Sass	3
Variables	3
Map	3
Mixins	4
Utilities API	4

Background color

Similar to the contextual text color classes, set the background of an element to any contextual class. Background utilities **do not set color**, so in some cases you'll want to use `.text-*` [color utilities]({{< docsref "/utilities/colors" >}}).

```
{{< example >}} {{< colors.inline >}} {{- range (index $.Site.Data "theme-colors") }}
```

```
.bg-{{ .name }}
```

```
{{- end -}} {{< /colors.inline >}}
```

```
.bg-body
```

```
.bg-white
```

```
.bg-transparent
```

```
{{< /example >}}
```

Background gradient

By adding a `.bg-gradient` class, a linear gradient is added as background image to the backgrounds. This gradient starts with a semi-transparent white which fades out to the bottom.

Do you need a gradient in your custom CSS? Just add `background-image: var(--bs-gradient);`.

```
{{< markdown >}} {{< colors.inline >}} {{- range (index $.Site.Data "theme-  
colors") }}  
  
.bg-{{ .name }}.bg-gradient  
{{- end -}} {{< /colors.inline >}} {{< /markdown >}}
```

Opacity

Added in v5.1.0

As of v5.1.0, `background-color` utilities are generated with Sass using CSS variables. This allows for real-time color changes without compilation and dynamic alpha transparency changes.

How it works

Consider our default `.bg-success` utility.

```
.bg-success {  
  --bs-bg-opacity: 1;  
  background-color: rgba(var(--bs-success-rgb), var(--bs-bg-opacity)) !important;  
}
```

We use an RGB version of our `--bs-success` (with the value of 25, 135, 84) CSS variable and attached a second CSS variable, `--bs-bg-opacity`, for the alpha transparency (with a default value 1 thanks to a local CSS variable). That means anytime you use `.bg-success` now, your computed `color` value is `rgba(25, 135, 84, 1)`. The local CSS variable inside each `.bg-*` class avoids inheritance issues so nested instances of the utilities don't automatically have a modified alpha transparency.

Example

To change that opacity, override `--bs-bg-opacity` via custom styles or inline styles.

```
{{< example >}}
```

This is default success background

This is 50% opacity success background

```
{{< /example >}}
```

Or, choose from any of the `.bg-opacity` utilities:

```
{{< example >}}
```

This is default success background

This is 75% opacity success background

This is 50% opacity success background

This is 25% opacity success background

This is 10% opacity success background

{{< /example >}}

Sass

In addition to the following Sass functionality, consider reading about our included [CSS custom properties]({{< docsref "/customize/css-variables" >}}) (aka CSS variables) for colors and more.

Variables

Most **background-color** utilities are generated by our theme colors, reassigned from our generic color palette variables.

{{< scss-docs name="color-variables" file="scss/_variables.scss" >}}

{{< scss-docs name="theme-color-variables" file="scss/_variables.scss" >}}

{{< scss-docs name="variable-gradient" file="scss/_variables.scss" >}}

Grayscale colors are also available, but only a subset are used to generate any utilities.

{{< scss-docs name="gray-color-variables" file="scss/_variables.scss" >}}

Map

Theme colors are then put into a Sass map so we can loop over them to generate our utilities, component modifiers, and more.

{{< scss-docs name="theme-colors-map" file="scss/_variables.scss" >}}

Grayscale colors are also available as a Sass map. **This map is not used to generate any utilities.**

{{< scss-docs name="gray-colors-map" file="scss/_variables.scss" >}}

RGB colors are generated from a separate Sass map:

{{< scss-docs name="theme-colors-rgb" file="scss/_variables.scss" >}}

And background color opacities build on that with their own map that's consumed by the utilities API:

{{< scss-docs name="utilities-bg-colors" file="scss/_variables.scss" >}}

Mixins

No mixins are used to generate our background utilities, but we do have some additional mixins for other situations where you'd like to create your own gradients.

```
{{< scss-docs name="gradient-bg-mixin" file="scss/mixins/_gradients.scss" >}}  
{{< scss-docs name="gradient-mixins" file="scss/mixins/_gradients.scss" >}}
```

Utilities API

Background utilities are declared in our utilities API in `scss/_utilities.scss`.
[Learn how to use the utilities API.]({{< docsref "/utilities/api#using-the-api" >}})

```
{{< scss-docs name="utils-bg-color" file="scss/_utilities.scss" >}}
```