

Dark Mode

It's effortless to switch Dark Mode in Phoenix. You can enable Dark Mode by default or create a Dark/Light switch if you want. To set the default mode "Dark",you can also set system default theme in Phoenix. Please see the [configuration page](#).

Toggle Dark Mode

Toggling dark mode is very easy in Phoenix. You can toggle dark, light or auto mode by using checkbox, radio input, switch input and custom icon component.


Checkbox

☐ Dark mode


Switch Input

☒ Dark Mode

Custom icon




Dropdown



Radio button

☒ Light ☐ Dark ☐ Auto

 Copy Code

[View code](#)

If you are using gulp based workflow

Modify Dark colors using SCSS

You can find all the variables used to create the dark mode in `/src/scss/theme/root/_dark.scss` file. If you want to override a variable, copy that variable to your `/src/scss/user.scss` file and update it as you see fit. When you change the variable with scss, make sure that the gulp is running.

If you are not using gulp based workflow

Modify Dark colors using CSS

You can find all the CSS variables used to create the dark mode in `/public/assets/css/theme.css` file. Look for the class `:root.dark` and you will see all the available variables. If you want to override a variable, copy that variable to your `/public/assets/css/user.css` file and update it as you see fit.

```
/*-----
Theme Styles
-----*/

:root, [data-bs-theme="light"] {
  --phoenix-body-color: #31374a;
  --phoenix-body-color-rgb: 49, 55, 74;

  --phoenix-body-bg: #e3e6ed;
  --phoenix-body-bg-rgb: 245, 247, 250;
}

[data-bs-theme="dark"] {
  --phoenix-body-color: #9fa6bc;
  --phoenix-body-color-rgb: 159, 166, 188;

  --phoenix-body-bg: #0f111a;
  --phoenix-body-bg-rgb: 15, 17, 26;
}
```

Using the Dark CSS classes in HTML

You can keep a style constant regardless of current (light or dark) mode

If you want a component to retain it's color (light or dark) as it is regardless of the current mode, you can use the following classes -

`[data-bs-theme="light"]` - It will keep the color light even if the current mode is dark

`[data-bs-theme="dark"]` - It will keep the color dark even if the current mode is light

The following two examples illustrate the color persistency -

Keep the color light even if the current mode is dark

This element will retain its color if you switch between light and dark modes.

```
<div class="card bg-body-highlight" data-bs-theme="light">
  <div class="card-body">
    <p class="mb-0 text-body-tertiary"><b>This element will retain its color if you switch be
  </div>
</div>
```

Keep the color dark even if the current mode is light

This element will retain its color if you switch between light and dark modes.

```
<div class="card bg-body-highlight" data-bs-theme="dark">
  <div class="card-body">
    <p class="mb-0 text-body-tertiary"><b>This element will retain its color if you switch be
  </div>
</div>
```

Override Background and Text color only for dark mode

If you want to use a different text color or background color rather than the default dark theme color for any element, you can use the special "dark" classes:

- `dark__bg-*`
- `dark__text-*`

The following element illustrates the example:

This element will get different bg color rather than the default dark theme color.

```
<div class="card bg-light dark__bg-primary">
  <div class="card-body">
    <p class="mb-0"><span class="fw-bold">This element will get different bg color rather than the default dark theme color.</span></p>
  </div>
</div>
```

Emit JavaScript event on color scheme change

When you switch between the dark and light mode, or change any settings from the global theme config at runtime, we emit an event `clickControl`. We used this event to update colors using JavaScript. For example, the charts change their colors using this event. You can catch and use this event with the following code snippet:

```
const themeController = document.body;

themeController.addEventListener(
  "clickControl",
  ({ detail: { control, value } }) => {

    if (control === "phoenixTheme") {
      // value will be localStorage theme value (dark/light/auto)
      const mode = value === 'auto' ? window.phoenix.utils.getSystemTheme() : value;
      console.log(mode)
      // your code here
    }
  }
);
```

Thank you for creating with Phoenix | 2025 © Themewagon

v1.22.0