

Pure CSS dark-mode toggle

The simplest pure CSS toggle for dark mode without persistency. Commented example. Just read the code .

Contents

1	Code	1
2	„Screenshot“ :-)	2
3	Appendix	3
3.1	Code with :has	3
3.2	Code with :root and variables for a dedicated colorscheme	3
3.3	Persistency with minimal Javascript	4

1 Code

```
<html><head>
  <style>
```

```
/* invert the #content, if an input with id dark is checked. See
   https://developer.mozilla.org/en-US/docs/Web/CSS/Subsequent-sibling_
   ↪ combinator
   ,*/
input#dark:checked ~ #content {
  filter: invert(100%);
}
/* don't show the actual checkbox */
input#dark {
  display: none;
}
/* wrapper for the actual content, so we can make it depend on the
   checkbox.*/
#content {
  background-color: yellow;
```

```

    margin: 0px;
    padding: 0.5em;
}
/* label to toggle the checkbox without having to have the
   checkbox. */
#dark-label {
    position: absolute;
    background-color: red;
    /* keep the label on top of content if the style of content
       changes (on click of the dark mode). DANGER: this can become a
       maintenance nightmare. */
    z-index: 1;
}
/* don't have a margin around the body, so #content seems to be the full
   ↪ page.*/
body {
    margin: 0px;
}

```

```

</style>
</head><body>
  <!-- add the actual checkbox, hidden with display=none -->
  <input id="dark" name="dark" type="checkbox" value=""/>
  <!-- add a label for the checkbox. the for="dark" makes
       it check the checkbox with id dark. -->
  <label id="dark-label" for="dark">dark</label>
  <!-- the content wrapper, affected by the state of the
       checkbox with the same parent (parent is the <body> tag) -->
  <div id="content"><br>foo</div>
</body></html>

```

2 „Screenshot“ :-)

For technical reasons the „screenshot“ is HTML only.

This used to be a dark art, but the sibling combinator makes it clean and standard.

It can be used for a lot more interactivity than a dark-mode, but this is a beautifully simple example.

3 Appendix

3.1 Code with :has

The code can be simplified with modern CSS features, but this requires [Firefox 121 or later*](#), and [8.6% of FF users still use FF 115](#), so it would need a polyfill (and I don't minimize my use of JS to only where it's really needed¹ to then [pull in 191kB of polyfill](#)).

Give it another half decade or so to mature and actually get into the 99.9% of used infrastructure ().

But anyway: [thanks to Artyom Bologov](#) for the tip!

```
body:has(#dark:checked) {  
  filter: invert(100%);  
}
```

3.2 Code with :root and variables for a dedicated colorscheme

Instead of inverting, you can also use a separate colorscheme for darkmode which can look much cleaner (your designers will request that).

This requires much more work than faking a dark mode with a filter.

And using variables defined at :root in other places would require **hacks to keep working in simpler browsers** like dillo and eww and KaiOS.

Also that's newfangled stuff (), and it uses **more global state** (the ID in for="..." is global state too; but can be any random string, because it does not carry its own semantics), so while it's a clean solution, it's not what I currently prefer.

Thanks to Artyom Bologov [for the tip!](#)

```
:root:has(#dark:checked) {  
  --bg-color: black;  
}  
:root:has(#dark:not(:checked)) {  
  --bg-color: white;  
}  
body {  
  background-color: var(--bg-color);  
}  
/* and a lot more color definitions. */
```

¹Following the [Zen for Scheme](#) WM: "Use the Weakest Method that gets the job done, but know the stronger methods to employ them as needed."

3.3 Persistency with minimal Javascript

I don't think you can have a persistent trigger in pure CSS.

So **persistency requires Javascript** (or server-side state).

But you can have a minimum of Javascript that uses local storage for setting the checkbox.

If you did not click the dark-mode button, it follows your browser preferences, and once you toggle it manually, it records your preference in local storage.

```
// @license magnet:?xt=urn:btih:1f739d935676111cfff4b4693e3816e66479705_
↪ 0&dn=gpl-3.0.txt GPL-v3-or-Later
function invert(record = true) {
  if (!window.invertCounter) {
    window.invertCounter = 1
  } else {
    window.invertCounter++
  }
  const mustInvert = window.invertCounter % 2 == 1;
  // css adjustment logic (i.e. toggle a checkbox)
  // ...
  // persistence logic
  if (record) {
    window.localStorage.setItem("draketo-dark", mustInvert);
  }
}
const draketoDark = window.localStorage.getItem("draketo-dark");
if (draketoDark === "true" ||
    (draketoDark !== "false"
     && window?.matchMedia?.('(prefers-color-scheme:dark)')?.matches)) {
  invert(record = false);
}
// @license-end
```

Example button to toggle dark-mode (could be the checkbox).

```
<label for="css-dark-mode-toggle-checkbox">(dark mode) </span>
<input id="css-dark-mode-toggle-checkbox" type="checkbox"
↪  oninput="invert()"></input>
<script>
const draketoDarkMode = window.localStorage.getItem("draketo-dark");
if (draketoDarkMode === "true" ||
    (draketoDarkMode !== "false"
     && window?.matchMedia?.('(prefers-color-scheme:dark)')?.matches)) {
```

```
document.getElementById("css-dark-mode-toggle-checkbox").checked =  
  ↪ true;  
}  
</script>
```

To test: