FED 24/25
Kleurtjes met
has en lightdark
intro







Support voor :has()



De grote browsers werken samen om elk jaar tegelijk op dezelfde manier nieuwe technieken te introduceren: wpt.fyi/interop-2024

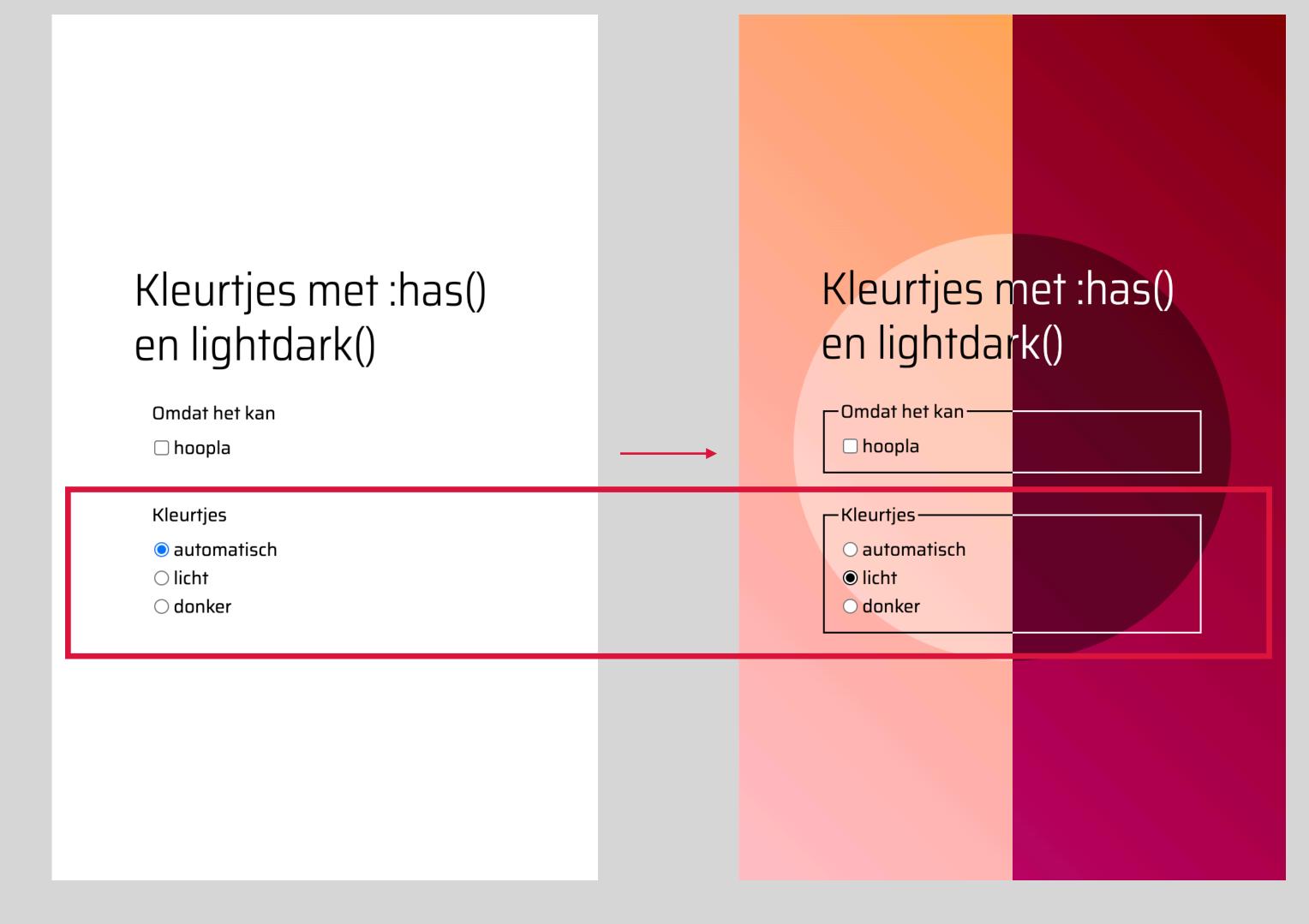
De actuele support voor :has() op caniuse: caniuse.com/css-has → 92.5% (not bad :)

Support voor : light-dark()



Om auto, light en dark mode te implementeren was altijd best veel JS nodig.

Met de relatief nieuwe :has() en light-dark() features kan dat met veel minder code en alleen met CSS.

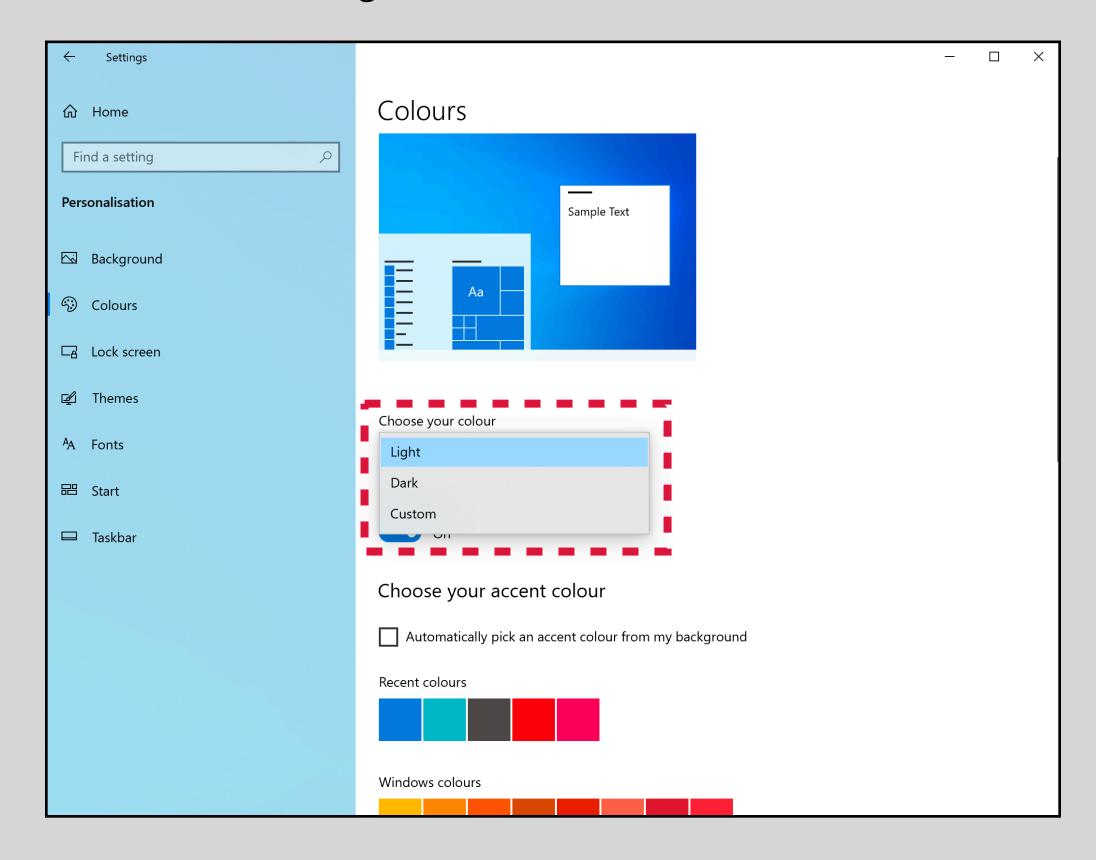


start: codepen.io/shooft/pen/mdZjZNW

auto, licht en donker: codepen.io/shooft/pen/xxozyVW

Dark/Light - Windows

Settings → Personalisation → Colours



Dark/Light - macOS

System preferences → general

• • < > :::: Gene	ral	Q Search
Appearance: Accent colour:	Light Dark Auto Graphite	
Highlight colour:	Yellow	
Sidebar icon size:	Medium	
	Allow wallpaper tinting in windows	
Show scroll bars:	Automatically based on mouse or toWhen scrollingAlways	ackpad
Click in the scroll bar to:	Jump to the next pageJump to the spot that's clicked	
Default web browser:	© Google Chrome	
Prefer tabs:	in full screen 🕞 when opening docu	iments
	Ask to keep changes when closing	documents
	Close windows when quitting an app When selected, open documents and win when you re-open an app.	
Recent items:	10 Ocuments, Apps and Serv	/ers
	Allow Handoff between this Mac an	d your iCloud devices
		?

Om auto, light en dark mode te implementeren was altijd best veel JS nodig.

Met de relatief nieuwe :has() en light-dark() features kan dat met veel minder code en alleen met CSS.

```
In de HTML
<label>
  <input type="radio" name="theme" value="auto" checked> auto
</label>
<label>
   <input type="radio" name="theme" value="light"> licht
</label>
<label>
  <input type="radio" name="theme" value="dark"> donker
</label>
In de CSS
:root {
  color-scheme: light dark;
  --bg: light-dark(Pink, MediumVioletRed);
  --clr: light-dark(Black, White);
html:has([value="light"]:checked) {
   color-scheme: light;
html:has([value="dark"]:checked) {
   color-scheme: dark;
html {
  color: var(--clr);
  background-color: var(--bg);
```

Drie radio buttons

voor auto (default),

light en dark

Om auto, light en dark mode te implementeren was altijd best veel JS nodig.

Met de relatief nieuwe :has() en light-dark() features kan dat met veel minder code en alleen met CSS.

In de HTML

In de CSS

```
:root {
    color-scheme: light dark;

    --bg: light-dark(Pink, MediumVioletRed);
    --clr: light-dark(Black, White);
}

html:has([value="light"]:checked) {
    color-scheme: light;
}

html:has([value="dark"]:checked) {
    color-scheme: dark;
}

html {
    color: var(--clr);
```

background-color: var(--bg);

Naast de standaard light modus van de browser ook de dark modus activeren.

De browser gebruikt dan de modus (light of dark) die actief is in het besturingssysteem.

Om auto, light en dark mode te implementeren was altijd best veel JS nodig.

Met de relatief nieuwe :has() en light-dark() features kan dat met veel minder code en alleen met CSS.

```
In de HTML
<label>
  <input type="radio" name="theme" value="auto" checked> auto
</label>
<label>
   <input type="radio" name="theme" value="light"> licht
</label>
<label>
  <input type="radio" name="theme" value="dark"> donker
</label>
In de CSS
:root {
                                                     Custom properties definiëren
                                                     met de light-dark() functie.
  color-scheme: light dark;
                                                     De custom properties hebben
  --bg: light-dark(Pink, MediumVioletRed);
                                                     dan een light en dark variant
  --clr: light-dark(Black, White);
                                                     die worden gebruikt als de light
                                                     of dark modus van de browser
                                                     actief is.
html:has([value="light"]:checked) {
   color-scheme: light;
html:has([value="dark"]:checked) {
   color-scheme: dark;
```

html {

color: var(--clr);

background-color: var(--bg);

Om auto, light en dark mode te implementeren was altijd best veel JS nodig.

Met de relatief nieuwe :has() en light-dark() features kan dat met veel minder code en alleen met CSS.

In de HTML

```
<label>
  <input type="radio" name="theme" value="auto" checked> auto
</label>
<label>
   <input type="radio" name="theme" value="light"> licht
</label>
<label>
  <input type="radio" name="theme" value="dark"> donker
</label>
In de CSS
:root {
  color-scheme: light dark;
  --bg: light-dark(Pink, MediumVioletRed);
  --clr: light-dark(Black, White);
html:has([value="light"]:checked) {
   color-scheme: light;
html:has([value="dark"]:checked) {
   color-scheme: dark;
                                                   De custom properties
                                                   gebruiken zoals altijd.
html {
  color: var(--clr);
```

background-color: var(--bg);

Om auto, light en dark mode te implementeren was altijd best veel JS nodig.

Met de relatief nieuwe :has() en light-dark() features kan dat met veel minder code en alleen met CSS.

```
In de HTML
<label>
 <input type="radio" name="theme" value="auto" checked> auto
</label>
<label>
   <input type="radio" name="theme" value="light"> licht
</label>
<label>
 <input type="radio" name="theme" value="dark"> donker
</label>
In de CSS
:root {
 color-scheme: light dark;
```

```
--bg: light-dark(Pink, MediumVioletRed);
  --clr: light-dark(Black, White);
:root:has([value="light"];checked) {
   color-scheme: light;
:root:has([value="dark"]:checked) {
   color-scheme: dark;
html {
  color: var(--clr);
  background-color: var(--bg);
```

Als de radio button met de value light gecheckt is, mag de browser alleen de light modus gebruiken.

Als de radio button met de value dark gecheckt is, mag de browser alleen de dark modus gebruiken.



FED 24/25 Kleurtjes met has en lightdark intro