

CSS challenge #6

LIGHTDARK  
INTRO

# Deel 1.

Custom properties &  
light dark

**color-scheme**

Custom properties

@media

light-dark()

:has()

Oefenen

# Deel 2.

Icons

SVG en IMG

# color-scheme



A screenshot of a web form titled 'Een formulier' (A form) in a light theme. The form is contained within a white rounded rectangle with a black border. It includes a title, a descriptive paragraph, two input fields for 'je e-mailadres' and 'je wachtwoord', and a submit button labeled 'ok dan'.

**Een formulier**

Een formulier is een middel om gestructureerd informatie (een verzoek, vaststelling, verklaring, beschikking, enz.) vast te leggen.

je e-mailadres

je wachtwoord

Van oudsher had een browser alleen een lichte style sheet, maar tegenwoordig heeft bijna elke browser er 2. een lichte en een donkere.

De donkere stylesheet staat alleen standaard niet 'aan'. Dan zouden veel oude websites 'stuk' gaan.



A screenshot of the same web form as on the left, but rendered in a dark theme. The background is black, and the form container is a dark rounded rectangle with a white border. The text and input fields are white, providing high contrast.

**Een formulier**

Een formulier is een middel om gestructureerd informatie (een verzoek, vaststelling, verklaring, beschikking, enz.) vast te leggen.

je e-mailadres

je wachtwoord

Om de donkere stylesheet ook 'aan' te zetten, gebruik je de color-scheme property:

```
html {  
    color-scheme: light dark;  
}
```

# Deel 1.

Custom properties &  
light dark

~~color-scheme~~

**Custom properties**

@media

light-dark()

:has()

Oefenen

# Deel 2.

Icons

SVG en IMG

# CSS custom properties



Ik wil de H1  
HotPink maken

Vandaag gebruiken we **custom properties voor kleuren** - een prima start. Je kunt ze echter voor veel meer gebruiken: fonts, border-radius, white-space...

En dan kun je ook nog met ze rekenen, kun je ze met JS aanpassen, en kun je ze met een beetje moeite zelfs animeren. Je gaat ze zeker nog vaker tegenkomen.

**Dat kan natuurlijk zo:**

```
h1 {  
  color: HotPink;  
}
```

**Maar dit is beter:**

```
:root {  
  --merk-kleur: HotPink;  
}
```

Dat is een CSS  
custom property.

```
h1 {  
  color: var(--merk-kleur);  
}
```

Die kun je vervolgens  
gebruiken als waarde voor  
een 'normale' property.

Dat is nog niet zo  
interessant. Dat wordt het  
wel als je de waarde op  
meer plekken nodig hebt.

# CSS custom properties

Vandaag gebruiken we **custom properties voor kleuren** - een prima start. Je kunt ze echter voor veel meer gebruiken: fonts, border-radius, white-space...

En dan kun je ook nog met ze rekenen, kun je ze met JS aanpassen, en kun je ze met een beetje moeite zelfs animeren. Je gaat ze zeker nog vaker tegenkomen.

Annuleren

OK

Nu wordt het interessanter:

```
:root {  
  --merk-kleur: HotPink;  
}  
  
h1 {  
  color: var(--merk-kleur);  
}  
  
button {  
  color: var(--merk-kleur);  
  background-color: White;  
  border: solid 4px var(--merk-kleur);  
}  
  
button[type="submit"] {  
  color: White;  
  background-color: var(--merk-kleur);  
}
```

# CSS custom properties

Vandaag gebruiken we **custom properties voor kleuren** - een prima start. Je kunt ze echter voor veel meer gebruiken: fonts, border-radius, white-space...

En dan kun je ook nog met ze rekenen, kun je ze met JS aanpassen, en kun je ze met een beetje moeite zelfs animeren. Je gaat ze zeker nog vaker tegenkomen.

Annuleren

OK

code: [start uitwerking](#)

Nu wordt het interessanter:

Één waarde  
veranderen en alles  
is aangepast.

```
:root {  
  --merk-kleur: RebeccaPurple;  
}  
  
h1 {  
  color: var(--merk-kleur);  
}  
  
button {  
  color: var(--merk-kleur);  
  background-color: White;  
  border: solid 4px var(--merk-kleur);  
}  
  
button[type="submit"] {  
  color: White;  
  background-color: var(--merk-kleur);  
}
```

# Anatomie van een CSS custom property

The diagram illustrates the anatomy of a CSS custom property through two code snippets. The first snippet defines the property in the root, and the second snippet uses it in a selector. Red arrows point from explanatory text to specific parts of the code.

```
:root {  
  --merk-kleur: HotPink;  
}  
  
h1 {  
  color: var(--merk-kleur);  
}
```

Annotations:

- Definiëren in :root zodat ze in je hele CSS bestand beschikbaar zijn (points to `:root`)
- Beginnen met 2 streepjes (points to `--`)
- Een robuuste betekenisvolle naam (points to `merk-kleur`)
- De waarde die je de custom property wilt geven (points to `HotPink`)
- Aanroepen met `var()` (points to `var()`)
- Met daarin de naam van de custom property (points to `--merk-kleur`)



# Deel 1.

Custom properties &  
light dark

~~color-scheme~~

~~Custom-properties~~

**@media**

light-dark()

:has()

Oefenen

# Deel 2.

Icons

SVG en IMG

# CSS custom properties

Annuleren

OK

# CSS custom properties

Annuleren

OK

## Hoepla een Dark theme

```
:root {
  --merk-kleur: HotPink;
  --button-achtergrond: White;
  --submit-kleur: White;
}

@media (prefers-color-scheme:dark) {
  :root {
    --merk-kleur: Lime;
    --button-achtergrond: Black;
    --submit-kleur: Black;
  }
}

h1 {
  color: var(--merk-kleur);
}

button {
  color: var(--merk-kleur);
  background-color: var(--button-achtergrond);
  border: solid 4px var(--merk-kleur);
}

button[type="submit"] {
  color: var(--submit-kleur: White);
  background-color: var(--merk-kleur);
```

# Deel 1.

Custom properties &  
light dark

~~color-scheme~~

~~Custom-properties~~

~~@media~~

**light-dark()**

:has()

Oefenen

# Deel 2.

Icons

SVG en IMG

# CSS custom properties

Annuleren

OK

# CSS custom properties

Annuleren

OK

De nieuwe manier om een lichte en een donkere kleur voor een custom property te definiëren is met de `light-dark()` function. Om die te laten werken moet je ook de `color-scheme` property gebruiken.

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

  --merk-clr: light-dark(HotPink, Lime);
  --button-bg: light-dark(#fff, #000);
  --submit-clr: light-dark(#fff, #000);
}

h1 {
  color: var(--merk-kleur);
}

button {
  color: var(--merk-kleur);
  background-color: var(--button-achtergrond);
  border: solid 4px var(--merk-kleur);
}

button[type="submit"] {
  color: var(--submit-kleur: White);
  background-color: var(--merk-kleur);
}
```

# Deel 1.

Custom properties &  
light dark

~~color-scheme~~

~~Custom-properties~~

~~@media~~

~~light-dark()~~

**:has()**

Oefenen

# Deel 2.

Icons

SVG en IMG

advanced

Om de color-theme setting van je computer in je website te overrulen kun je in je html 3 radio buttons opnemen:

```
<label>
  <input type="radio" name="color-scheme" value="auto" checked>
  auto
</label>
<label>
  <input type="radio" name="color-scheme" value="light"> licht
</label>
<label>
  <input type="radio" name="color-scheme" value="dark"> donker
</label>
```

In de css kun je dan met :has() aangeven dat maar één van beide color-scheme gebruikt moet worden:

```
:root {
  color-scheme: light dark;
  --theme-color: light-dark(#609, #c9f);
  --tekst-color: light-dark(#069, #9cf);
}
```

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

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



● auto ○ licht ○ donker

## Een formulier

Een formulier is een middel om gestructureerd informatie (een verzoek, vaststelling, verklaring, beschikking, enz.) vast te leggen.

je e-mailadres

je wachtwoord

☐ ok dan



○ auto ○ licht ● donker

## Een formulier

Een formulier is een middel om gestructureerd informatie (een verzoek, vaststelling, verklaring, beschikking, enz.) vast te leggen.

je e-mailadres

je wachtwoord

☐ ok dan

# Deel 1.

Kleurtjes

Syntax

Mooie kleuren

Gradients

Kleuren mengen

Oefenen

# Deel 2.

Custom properties &  
light dark

color-scheme

Custom properties

@media

light-dark()

:has()

**Oefenen**

# Deel 3.

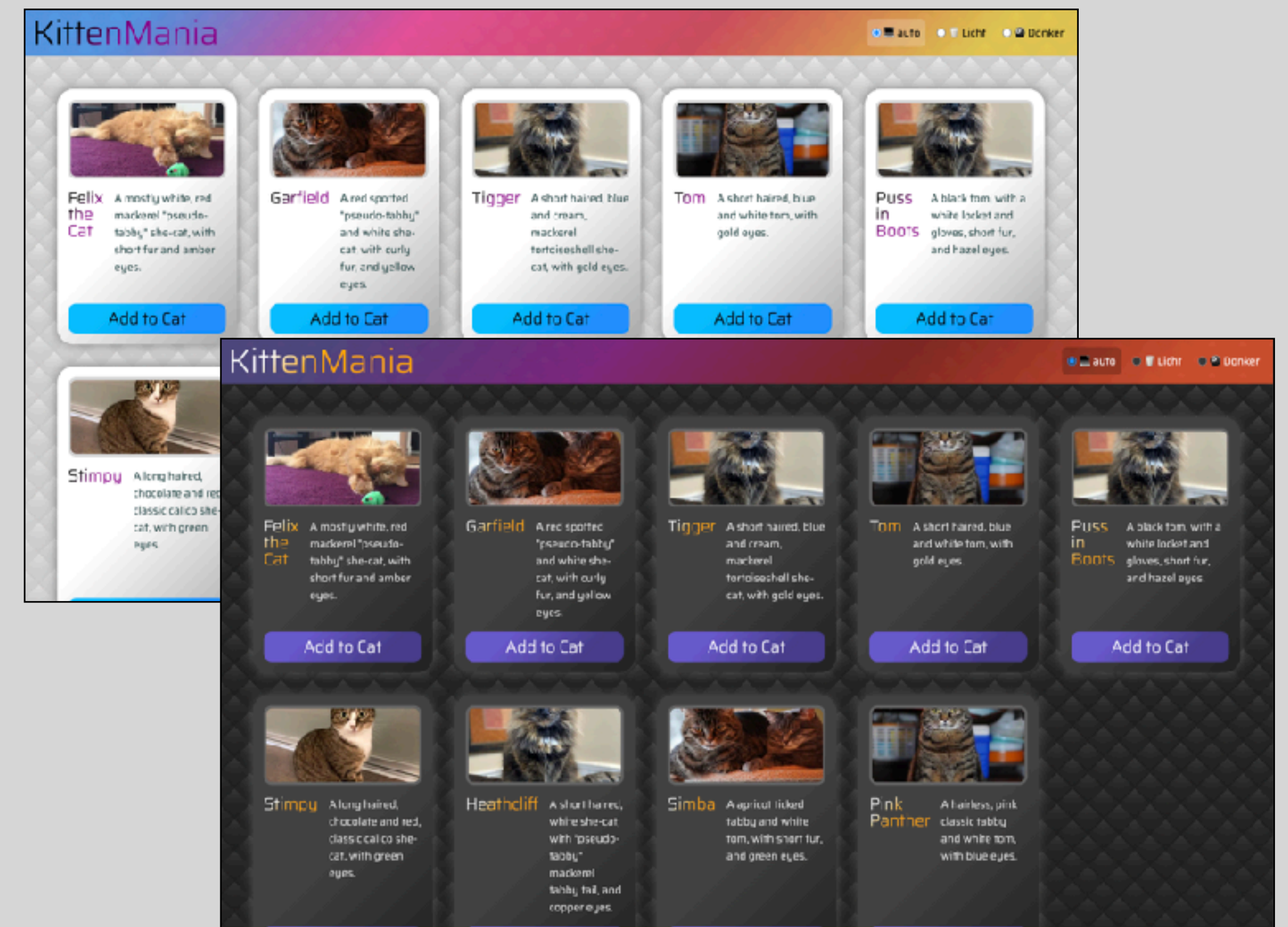
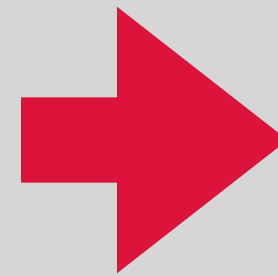
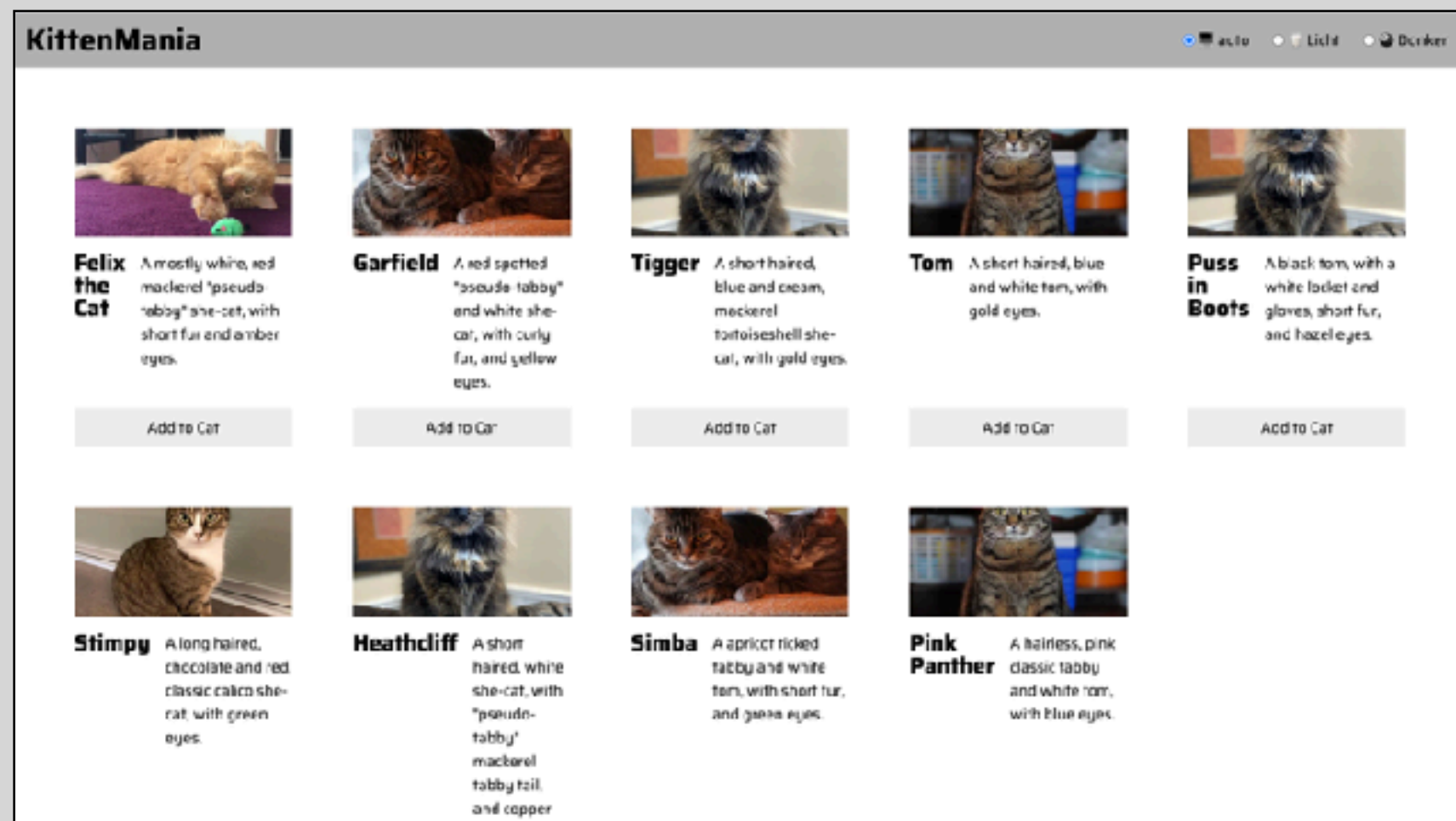
Icons

SVG en IMG



# Oefening

Maak mooie (of heel lelijk) kleurenthema's voor de kattenwinkel.



Code voor jou:

[codepen.io/shoof/pen/RNrLwmq](https://codepen.io/shoof/pen/RNrLwmq)

Een expres lelijk voorbeeld :)

[codepen.io/shoof/live/wBMrvGQ](https://codepen.io/shoof/live/wBMrvGQ)



# Deel 1.

Custom properties &  
light dark

~~color-scheme~~

~~Custom-properties~~

~~@media~~

~~light-dark()~~

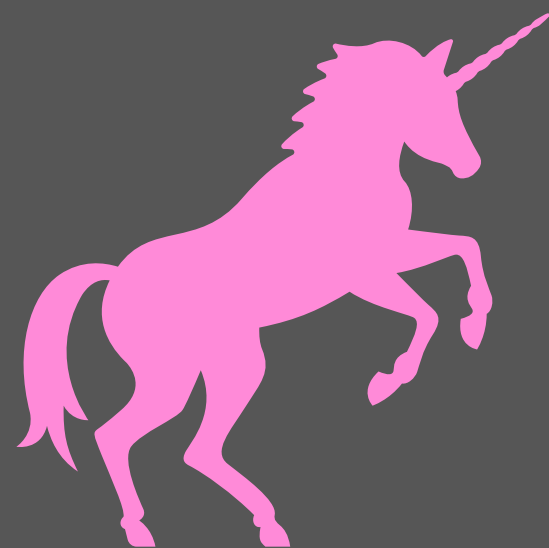
~~:has()~~

~~Oefenen~~

# Deel 2.

Icons

**SVG en IMG**

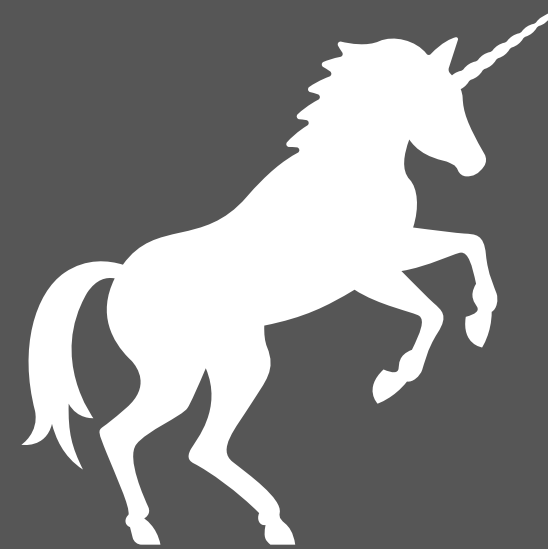
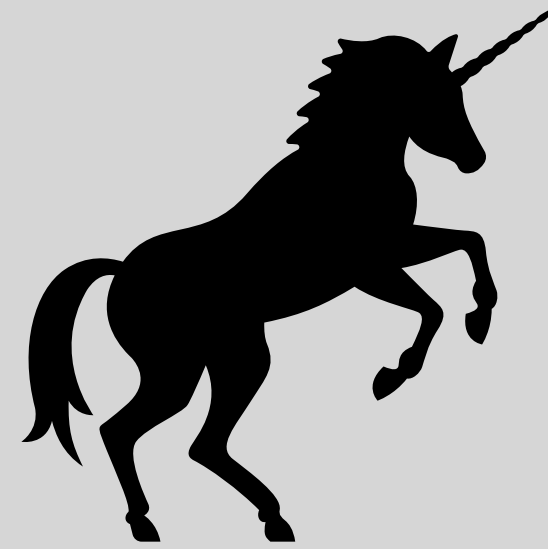


```
<svg viewBox="0 0 64 64">
  <path d="M14.2,48.3c1.6-1.7Z"/>
</svg>
```

```
svg {
  fill:purple;
}
```

```
html {
  --icon-color:light-dark(Purple, Pink);
}
```

```
svg {
  fill: var(--icon-color);
}
```



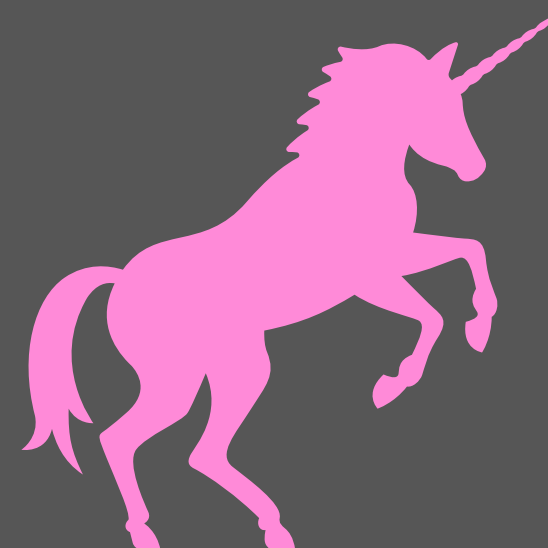
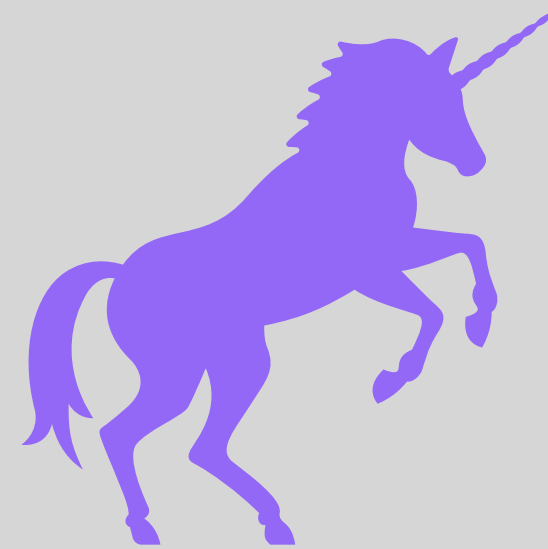
**Als je een icon als img in je html opneemt:**

```

```

**Dan kun je deze met een filter van zwart naar wit veranderen:**

```
@media (prefers-color-scheme:dark) {  
  svg{  
    filter:invert(100%);  
  }  
}
```



Als je een icon als img in je html opneemt:

```

```

Dan kun je deze met een complex filter van de ene kleur naar andere kleur veranderen.

1. Eerst maak je het icon zwart:

```
@media (prefers-color-scheme:dark) {  
  {  
    filter: brightness(0);  
  }  
}
```

2. Daarna kun je de rest van het filter genereren met:  
<https://codepen.io/sosuke/full/Pjoqqp>:

```
@media (prefers-color-scheme:dark) {  
  {  
    filter: brightness(0) invert(82%) sepia(11%)  
    saturate(3871%) hue-rotate(282deg)  
    brightness(100%) contrast(102%);  
  }  
}
```

# INITBO GICHTDVAVK

с22 challenge #e