



# filtre

La propriété [CSS](#) applique des effets graphiques comme le flou ou le changement de couleur à un élément. Les filtres sont couramment utilisés pour ajuster le rendu des images, des arrière-plans et des bordures. **filter**

La norme CSS comprend plusieurs fonctions qui permettent d'obtenir des effets prédéfinis. Vous pouvez également référencer un filtre SVG avec une URL vers un [élément de filtre SVG](#).

CSS Demo: filter

filter: url("../media


filter: blur(5px);

filter: contrast(200%);

filter: grayscale(80%);

filter: hue-rotate(90deg

Reset



## Syntaxe

```
/* URL to SVG filter */
filter: url("filters.svg#filter-id");

/* <filter-function> values */
filter: blur(5px);
filter: brightness(0.4);
filter: contrast(200%);
filter: drop-shadow(16px 16px 20px blue);
filter: grayscale(50%);
filter: hue-rotate(90deg);
filter: invert(75%);
```



```
filter: opacity(25%);
filter: saturate(30%);
filter: sepia(60%);

/* Multiple filters */
filter: contrast(175%) brightness(3%);

/* Use no filter */
filter: none;

/* Global values */
filter: inherit;
filter: initial;
filter: revert;
filter: unset;
```

Avec une fonction, utilisez ce qui suit :

```
filter: <filter-function> [<filter-function>]* | none
```



Pour une référence à un [<filter>](#) élément SVG, utilisez ce qui suit :

```
filter: url(file.svg#filter-element-id)
```



## Interpolation

Si les filtres de début et de fin ont tous deux une liste de fonctions de même longueur sans [url\(\)](#), chacune de leurs fonctions de filtre est interpolée selon ses règles spécifiques. Si elles ont des longueurs différentes, les fonctions de filtre équivalentes manquantes de la liste la plus longue sont ajoutées à la fin de la liste la plus courte en utilisant leurs valeurs de lacune, puis toutes les fonctions de filtre sont interpolées selon leurs règles spécifiques. Si un filtre est `none`, il est remplacé par la liste des fonctions de filtrage de l'autre en utilisant les valeurs par défaut de la fonction de filtrage, puis toutes les fonctions de filtrage sont interpolées selon leurs règles spécifiques. Sinon, une interpolation discrète est utilisée.

## Les fonctions

La `filter` propriété est spécifiée en tant `none` qu'une ou plusieurs des fonctions répertoriées ci-dessous. Si le paramètre d'une fonction n'est pas valide, la fonction renvoie `none`. Sauf indication contraire, les fonctions qui prennent une valeur exprimée avec un signe de pourcentage (comme dans `34%`) acceptent également la valeur exprimée sous forme décimale (comme dans `0.34`).

Lorsqu'une seule `filter` propriété a deux fonctions ou plus, ses résultats seront différents de

ceux obtenus lorsque deux `filter` propriétés ou plus sont appliquées séparément avec les mêmes fonctions.

## Filtre SVG

### URL()

Prend un URI pointant vers un [filtre SVG](#) , qui peut être intégré dans un fichier XML externe.

```
filter: url(resources.svg#c1)
```



## Fonctions de filtrage





### se brouiller()

La [blur\(\)](#) fonction applique un flou gaussien à l'image d'entrée. La valeur de `radius` définit la valeur de l'écart type de la fonction gaussienne, ou le nombre de pixels sur l'écran qui se fondent les uns dans les autres, donc une valeur plus grande créera plus de flou. La valeur de la lacune pour l'interpolation est `0` . Le paramètre est spécifié en tant que longueur CSS, mais n'accepte pas les valeurs en pourcentage.

```
filter: blur(5px)
```



```
<svg style="position: absolute; top: -99999px" xmlns="http://www.w3.org/2000/svg">
  <filter id="svgBlur" x="-5%" y="-5%" width="110%" height="110%">
    <feGaussianBlur in="SourceGraphic" stdDeviation="5"/>
  </filter>
</svg>
```

Original image	Live example	SVG Equivalent	Static example
			

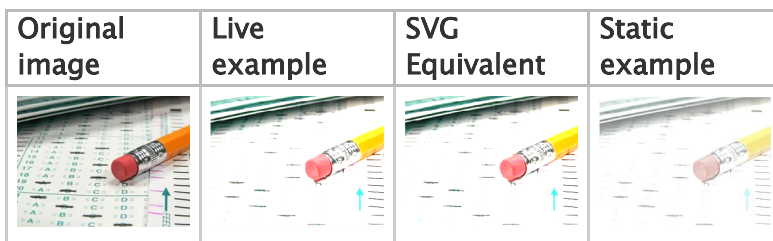
### luminosité()

La [brightness\(\)](#) fonction applique un multiplicateur linéaire à l'image d'entrée, la faisant apparaître plus ou moins lumineuse. Une valeur de `0%` créera une image complètement noire. Une valeur de `100%` laisse l'entrée inchangée. Les autres valeurs sont des multiplicateurs linéaires sur l'effet. Les valeurs supérieures à un montant `100%` sont autorisées, ce qui donne des résultats plus brillants. La valeur de la lacune pour l'interpolation est `1` .

**filter: brightness(2)**



```
<svg style="position: absolute; top: -99999px" xmlns="http://www.w3.org/2000/svg">
  <filter id="brightness">
    <feComponentTransfer>
      <feFuncR type="linear" slope="[amount]" />
      <feFuncG type="linear" slope="[amount]" />
      <feFuncB type="linear" slope="[amount]" />
    </feComponentTransfer>
  </filter>
</svg>
```







## contraste()

La [contrast\(\)](#) fonction ajuste le contraste de l'image d'entrée. Une valeur de 0% créera une image complètement grise. Une valeur de 100% laisse l'entrée inchangée. Les valeurs supérieures à un montant 100% sont autorisées, ce qui donne des résultats plus contrastés. La valeur de la lacune pour l'interpolation est 1.

**filter: contrast(200%)**



```
<svg style="position: absolute; top: -99999px" xmlns="http://www.w3.org/2000/svg">
  <filter id="contrast">
    <feComponentTransfer>
      <feFuncR type="linear" slope="[amount]" intercept="-(0.5 * [amount]) + 1" />
      <feFuncG type="linear" slope="[amount]" intercept="-(0.5 * [amount]) + 1" />
      <feFuncB type="linear" slope="[amount]" intercept="-(0.5 * [amount]) + 1" />
    </feComponentTransfer>
  </filter>
</svg>
```

Original image	Live example	SVG Equivalent	Static example
			

## ombre portée()

La [drop-shadow\(\)](#) fonction applique un effet d'ombre portée à l'image d'entrée. Une ombre portée est en fait une version floue et décalée du masque alpha de l'image d'entrée dessinée dans une couleur particulière, composée sous l'image. La fonction accepte un paramètre de type `<shadow>` (défini dans [Arrière-plans CSS3](#) ), à l'exception que le `inset` mot-clé et le `spread` paramètre ne sont pas autorisés. Cette fonction est similaire à la [box-shadow](#) propriété plus établie ; la différence est qu'avec les filtres, certains navigateurs fournissent une accélération matérielle pour de meilleures performances. Les paramètres du `<shadow>` paramètre sont les suivants :

### `<offset-x> <offset-y>` (obligatoire)

Ce sont deux [<length>](#) valeurs pour définir le décalage de l'ombre.

`<offset-x>` spécifie la distance horizontale. Les valeurs négatives placent l'ombre à gauche de l'élément. `<offset-y>` spécifie la distance verticale. Les valeurs négatives placent l'ombre au-dessus de l'élément. Voir [<length>](#) les unités possibles. Si les deux valeurs sont `0`, l'ombre est placée derrière l'élément (et peut générer un effet de flou si `<blur-radius>` et/ou `<spread-radius>` est défini).

### `<blur-radius>` (optionnel)

C'est une troisième [<length>](#) valeur. Plus cette valeur est élevée, plus le flou est important, donc l'ombre devient plus grande et plus claire. Les valeurs négatives ne sont pas autorisées. S'il n'est pas spécifié, il le sera `0` (le bord de l'ombre est net).

### `<color>` (optionnel)

Voir [<color>](#) les valeurs pour les mots-clés et les notations possibles. Si elle n'est pas spécifiée, la couleur utilisée dépend du navigateur - il s'agit généralement de la valeur de la [<color>](#) propriété, mais notez que Safari peint actuellement une ombre transparente dans ce cas.





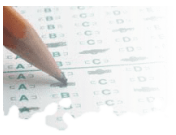


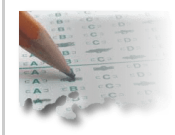
**filter:** `drop-shadow(16px 16px 10px black)`

```

<svg style="position: absolute; top: -999999px" xmlns="http://www.w3.org/2000/svg"
  <filter id="drop-shadow">
    <feGaussianBlur in="SourceAlpha" stdDeviation="[radius]"/>

    <feOffset dx="[offset-x]" dy="[offset-y]" result="offsetblur"/>
    <feFlood flood-color="[color]"/>
    <feComposite in2="offsetblur" operator="in"/>
    <feMerge>
      <feMergeNode/>
      <feMergeNode in="SourceGraphic"/>
    </feMerge>
  </filter>
</svg>

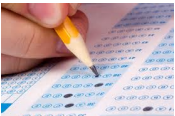



```

Original image	Live example	SVG Equivalent	Static example
			
			

niveaux de gris()

La [grayscale\(\)](#) fonction convertit l'image d'entrée en niveaux de gris. La valeur de amount définit la proportion de la conversion. Une valeur de 100% est complètement en niveaux de gris. Une valeur de 0% laisse l'entrée inchangée. Les valeurs entre 0% et 100% sont des multiplicateurs linéaires sur l'effet. La valeur de la lacune pour l'interpolation est 0.

**filter:** `grayscale(100%)`





Original image	Live example	SVG Equivalent	Static example
			

teinte-rotation()

La [hue-rotate\(\)](#) fonction applique une rotation de teinte sur l'image d'entrée. La valeur de angle définit le nombre de degrés autour du cercle de couleur où les échantillons d'entrée seront ajustés. Une valeur de 0deg laisse l'entrée inchangée. La valeur de la lacune pour l'interpolation est 0 . Bien qu'il n'y ait pas de valeur maximale; l'effet des valeurs ci-dessus 360deg s'enroule autour.

`filter: hue-rotate(90deg)`

```
<svg style="position: absolute; top: -999999px" xmlns="http://www.w3.org/2000/svg">
  <filter id="svgHueRotate">
    <feColorMatrix type="hueRotate" values="[angle]"/>
  </filter>
</svg>
```

Original image	Live example	SVG Equivalent	Static example
			





inverser()

La [invert\(\)](#) fonction inverse les échantillons dans l'image d'entrée. La valeur de amount définit la proportion de la conversion. Une valeur de 100% est complètement inversée. Une valeur de 0% laisse l'entrée inchangée. Les valeurs entre 0% et 100% sont des

multiplicateurs linéaires sur l'effet. La valeur de la lacune pour l'interpolation est 0 .

```
filter: invert(100%)
```




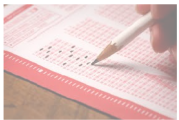
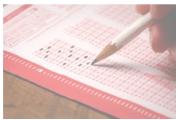
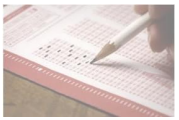
Original image	Live example	SVG Equivalent	Static example
			

opacité()

La [opacity\(\)](#) fonction applique la transparence aux échantillons dans l'image d'entrée. La valeur de amount définit la proportion de la conversion. Une valeur de 0% est complètement transparente. Une valeur de 100% laisse l'entrée inchangée. Les valeurs entre 0% et 100% sont des multiplicateurs linéaires sur l'effet. Cela équivaut à multiplier les échantillons d'image d'entrée par la quantité. La valeur de la lacune pour l'interpolation est 1 . Cette fonction est similaire à la [opacity](#) propriété plus établie ; la différence est qu'avec les filtres, certains navigateurs fournissent une accélération matérielle pour de meilleures performances.

```
filter: opacity(50%)
```



Original image	Live example	SVG Equivalent	Static example
			

saturer()



La [saturate\(\)](#) fonction sature l'image d'entrée. La valeur de `amount` définit la proportion de la conversion. Une valeur de `0%` est complètement insaturée. Une valeur de `100%` laisse l'entrée inchangée. Les autres valeurs sont des multiplicateurs linéaires sur l'effet. Des valeurs de quantité supérieures `100%` sont autorisées, fournissant des résultats sursaturés. La valeur de la lacune pour l'interpolation est `1`.

filter: saturate(200%)



Original image	Live example	SVG Equivalent	Static example

### sépia()

La [sepia\(\)](#) fonction convertit l'image d'entrée en sépia. La valeur de `amount` définit la proportion de la conversion. Une valeur de `100%` est complètement sépia. Une valeur de `0%` laisse l'entrée inchangée. Les valeurs entre `0%` et `100%` sont des multiplicateurs linéaires sur l'effet. La valeur de la lacune pour l'interpolation est `0`.

filter: sepia(100%)



Original image	Live example	SVG Equivalent	Static example

## Combinaison de fonctions

Vous pouvez combiner n'importe quel nombre de fonctions pour manipuler le rendu. L'exemple suivant améliore le contraste et la luminosité de l'image :

filter: contrast(175%) brightness(103%)



Original image	Live example	Static example

## Définition formelle

<b><u>Valeur initiale</u></b>
none
<b>S'applique à</b>
tous les éléments ; En SVG, il s'applique aux éléments conteneurs à l'exclusion de l' <a href="#">defs</a> élément et de tous les éléments graphiques
<b><u>Hérité</u></b>
non
<b><u>Valeur calculée</u></b>
comme spécifié
<b>Type d'animation</b>
une <b>liste de fonctions de filtrage</b>

## Syntaxe formelle

none | [.<filter-function-list>](#)  
where  
[<filter-function-list>](#) = [[<filter-function>](#) | [<url>](#)] 1+

<filter-function-list> = .[ <filter-function> .[ <filter-function> ]\*

where

<filter-function> = [<blur\(\)>](#) .| [<brightness\(\)>](#) .| [<contrast\(\)>](#) .| [<drop-shadow\(\)](#)

where

<blur()> = blur( [<length>](#) )

<brightness()> = brightness( [<number-percentage>](#) )

<contrast()> = contrast( .[ [<number-percentage>](#) ]\* )

<drop-shadow()> = drop-shadow( [<length>](#){2,3} [<color>?](#) )

<grayscale()> = grayscale( [<number-percentage>](#) )

<hue-rotate()> = hue-rotate( [<angle>](#) )

<invert()> = invert( [<number-percentage>](#) )

<opacity()> = opacity( .[ [<number-percentage>](#) ]\* )

<saturate()> = saturate( [<number-percentage>](#) )

<sepia()> = sepia( [<number-percentage>](#) )

where

<number-percentage> = [<number>](#) .| [<percentage>](#)

<color> = [<rgb\(\)>](#) .| [<rgba\(\)>](#) .| [<hsl\(\)>](#) .| [<hsla\(\)>](#) .| [<hwb\(\)>](#) .| <hex-color> .|

where

<rgb()> = rgb( [<percentage>](#){3} .[ / [<alpha-value>](#) ]? ) .| rgb( [<number>](#){3} .[ /

<rgba()> = rgba( [<percentage>](#){3} .[ / [<alpha-value>](#) ]? ) .| rgba( [<number>](#){3}

<hsl()> = hsl( [<hue>](#) [<percentage>](#) [<percentage>](#) .[ / [<alpha-value>](#) ]? ) .| hsl(

<hsla()> = hsla( [<hue>](#) [<percentage>](#) [<percentage>](#) .[ / [<alpha-value>](#) ]? ) .| hs

<hwb()> = hwb( .[[<hue>](#) .| none] .[[<percentage>](#) .| none] .[[<percentage>](#) .| none] .|

where

<alpha-value> = [<number>](#) .| [<percentage>](#)

<hue> = [<number>](#) .| [<angle>](#)

## Exemples

### Application des fonctions de filtrage

Des exemples d'utilisation des fonctions prédéfinies sont présentés ci-dessous. Voir chaque fonction pour un exemple spécifique.

```
.mydiv {
  filter: grayscale(50%);
}

/* Gray all images by 50% and blur by 10px */
img {
  filter: grayscale(0.5) blur(10px);
}
```



# Appliquer des filtres SVG

Voici des exemples d'utilisation de la fonction URL avec une ressource SVG :

```
.target {
  filter: url(#c1);
}

.mydiv {
  filter: url(commonfilters.xml#large-blur);
}
```



## Caractéristiques

spécification
<a href="#">Module d'effets de filtre Niveau 2 (Effets de filtre 2)</a> <a href="#"># FilterProperty</a>

## Compatibilité du navigateur

[Signaler les problèmes avec ces données de compatibilité sur GitHub](#)

filter	
Chrome	53
Edge	12
Firefox	35
Internet Explorer	Non
Opera	40
Safari	9.1
WebView Android	53
Chrome Android	53
Firefox for Android	35
Opera Android	41
Safari on iOS	9.3

Sur les éléments SVG

Chrome	Non
Edge	Non
Firefox	35
Internet Explorer	Non
Opera	Non
Safari	Non
WebView Android	Non
Chrome Android	Non
Firefox for Android	35
Opera Android	Non
Safari on iOS	Non
Samsung Internet	Non

☐

Plein soutien

☐

Pas de support

Voir les notes d'implémentation.

**-X-** Nécessite un préfixe de fournisseur ou un nom différent pour être utilisé.

Voir également

- [Application d'effets SVG au contenu HTML](#)
- La [mask](#) propriété
- [SVG](#)

**Dernière modification** : 18 janvier 2022, [par les contributeurs MDN](#)