



# Notation fonctionnelle CSS

La **notation fonctionnelle CSS** est un type de [valeur CSS](#) qui peut représenter des types de données plus complexes ou invoquer un traitement ou des calculs de données spéciaux.

## Syntaxe

```
selector {  
  property: functional-notation( [argument]? [, argument]! );  
}
```

La syntaxe commence par le **nom de la notation fonctionnelle**, suivi d'une parenthèse gauche ( . Viennent ensuite les arguments de notation et la fonction se termine par une parenthèse fermante ) .

Les fonctions peuvent prendre plusieurs arguments, qui sont formatés de la même manière que les valeurs de propriété CSS. Les espaces blancs sont autorisés, mais ils sont facultatifs à l'intérieur des parenthèses. Dans certaines notations fonctionnelles, plusieurs arguments sont séparés par des virgules, tandis que d'autres utilisent des espaces.

## Fonctions de transformation

Ces fonctions sont utilisées lorsque le [type de données <transform-function>](#) CSS est utilisé comme valeur de [.transform](#)

### [matrix\(\)](#)

Décrit une matrice de transformation 2D homogène.

### [matrix3d\(\)](#)

Décrit une transformation 3D comme une matrice homogène 4×4.

### [perspective\(\)](#)

Définit la distance entre l'utilisateur et le plan z=0.

### [rotate\(\)](#)

Fait pivoter un élément autour d'un point fixe sur le plan 2D.

### [rotate3d\(\)](#)

Fait pivoter un élément autour d'un axe fixe dans l'espace 3D.

### [rotatex\(\)](#)

Fait pivoter un élément autour de l'axe horizontal.

### [rotatey\(\)](#)

Fait pivoter un élément autour de l'axe vertical.

### [rotatez\(\)](#)

Fait pivoter un élément autour de l'axe z.

### [scale\(\)](#)

Met à l'échelle un élément vers le haut ou vers le bas sur le plan 2D.

### [scale3d\(\)](#)

Redimensionne un élément vers le haut ou vers le bas dans l'espace 3D.

### [scalex\(\)](#)

Redimensionne un élément vers le haut ou vers le bas horizontalement.

### [scaley\(\)](#)

Redimensionne un élément vers le haut ou vers le bas verticalement.

### [scalez\(\)](#)

Met à l'échelle un élément vers le haut ou vers le bas le long de l'axe z.

### [skew\(\)](#)

Incline un élément sur le plan 2D.

### [skewx\(\)](#)

Incline un élément dans le sens horizontal.

### [skewy\(\)](#)

Incline un élément dans le sens vertical.

### [translate\(\)](#)

Translate un élément sur le plan 2D.

### [translate3d\(\)](#)

Traduit un élément dans l'espace 3D.

### [translatex\(\)](#)

Traduit un élément horizontalement.

### [translatey\(\)](#)

Translate un élément verticalement.

### [translatez\(\)](#)

Translate un élément le long de l'axe z.

## Fonctions mathématiques

Les fonctions mathématiques permettent d'écrire des valeurs numériques CSS sous forme d'expressions mathématiques.

### [calc\(\)](#)

Une fonction mathématique qui permet d'effectuer des opérations arithmétiques de base sur des valeurs CSS numériques.

### [clamp\(\)](#)

Une fonction de comparaison qui prend une valeur minimale, centrale et maximale et représente son calcul central.

### [max\(\)](#)

Une fonction de comparaison qui représente la plus grande valeur d'une liste de valeurs.

### [min\(\)](#)

Une fonction de comparaison qui représente la plus petite valeur d'une liste de valeurs.

### [abs\(\)](#)

Effectue un calcul et renvoie la valeur absolue.

### [acos\(\)](#)

Une fonction trigonométrique inverse l'angle renvoyé doit être normalisé dans la plage [0deg, 180deg] ;.

### [asin\(\)](#)

Une fonction trigonométrique inverse l'angle renvoyé doit être normalisé à la plage [-90deg, 90deg].

### [atan\(\)](#)

Une fonction trigonométrique inverse l'angle renvoyé doit être normalisé à la plage [-90deg, 90deg].

### [atan2\(\)](#)

Contient deux calculs séparés par des virgules. A et B. A et B peut résoudre à tout

[<number>](#), [<dimension>](#) ou [<percentage>](#), mais ils doivent avoir le même type, ou bien la fonction est invalide.

### **cos()**

Contient un seul calcul qui doit être résolu en a [<number>](#) ou en [<angle>](#).

### **exp()**

Contient un calcul qui doit se résoudre en a [<number>](#) et renvoie la même valeur que pow(e, A) en tant que a [<number>](#).

### **hypot()**

Contient un ou plusieurs calculs séparés par des virgules et renvoie la longueur d'un vecteur à N dimensions avec des composants égaux à chacun des calculs.

### **log()**

Contient un ou deux calculs (représentant la valeur à logarithmer et la base du logarithme, par défaut e), qui doivent tous deux être résolus en tant que a [<number>](#), et renvoient la base du logarithme B de la valeur A, en tant que a [<number>](#).

### **mod()**

Une fonction de module qui contient deux calculs A et B et renvoie la différence entre A et le multiple entier le plus proche de B au-dessus ou au-dessous de A.

### **pow()**

Contient deux calculs séparés par des virgules A et B, qui doivent tous deux être résolus en tant que a [<number>](#), et renvoie le résultat de l'élévation de A à la puissance de B, renvoyant la valeur en tant que a [<number>](#).

### **rem()**

Une fonction de module qui contient deux calculs A et B et renvoie la différence entre A et le multiple entier le plus proche de B au-dessus ou au-dessous de A.

### **round()**

Contient une stratégie d'arrondi facultative et deux calculs A et B, et renvoie la valeur de A, arrondie selon la stratégie d'arrondi, au multiple entier le plus proche de B soit au-dessus soit au-dessous de A.

### **sign()**

Effectue un calcul et renvoie -1 si la valeur numérique est négative, +1 si la valeur numérique est positive, 0<sup>+</sup> si la valeur numérique est 0<sup>+</sup> et 0<sup>-</sup> si la valeur numérique est 0<sup>-</sup>.

### **sin()**

Contient un seul calcul qui doit être résolu en a [<number>](#) ou en [<angle>](#) .

**[sqrt\(\)](#)** 

Contient un seul calcul qui doit se résoudre en a [<number>](#) et renvoie la racine carrée de la valeur sous la forme a [<number>](#) .

**[tan\(\)](#)** 

Contient un seul calcul qui doit être résolu en a [<number>](#) ou en [<angle>](#) .

## Fonctions de filtrage

Le [type de données CSS](#) représente un effet graphique qui peut modifier l'apparence d'une image d'entrée. Il est utilisé dans les propriétés et . **<filter-function>**

[filter](#) [backdrop-filter](#)

**[blur\(\)](#)**

Brouille l'image.

**[brightness\(\)](#)**

Rend l'image plus lumineuse ou plus sombre.

**[contrast\(\)](#)**

Augmente ou diminue le contraste de l'image.

**[drop-shadow\(\)](#)**

Applique une ombre portée derrière l'image.

**[grayscale\(\)](#)**

Convertit l'image en niveaux de gris.

**[hue-rotate\(\)](#)**

Modifie la teinte générale de l'image.

**[invert\(\)](#)**

Inverse les couleurs de l'image.

**[opacity\(\)](#)**

Rend l'image transparente.

**[saturate\(\)](#)**

Super-sature ou désature l'image d'entrée.

## sepia()

Convertit l'image en sépia.

# Fonctions de couleur

Fonctions qui spécifient différentes représentations de couleurs. Ceux-ci peuvent être utilisés partout où a [<color>](#) est valide.

## color()

Permet à une couleur d'être spécifiée dans un espace colorimétrique particulier et spécifié (plutôt que l'espace colorimétrique sRGB implicite dans lequel la plupart des autres fonctions de couleur opèrent).

## color-mix()

Prend deux [color](#) valeurs et renvoie le résultat de leur mélange dans un espace colorimétrique donné d'une quantité donnée.

## color-contrast()

Prend une [color](#) valeur et la compare à une liste d'autres [color](#) valeurs, en sélectionnant celle qui présente le contraste le plus élevé dans la liste.

## device-cmyk()

Utilisé pour exprimer les couleurs CMJN d'une manière indépendante de l'appareil.

## hsl()

Le modèle de couleur HSL définit une couleur donnée en fonction de ses composantes de teinte, de saturation et de luminosité. Un composant alpha facultatif représente la transparence de la couleur.

## hsla()

Le modèle de couleur HSL définit une couleur donnée en fonction de ses composantes de teinte, de saturation et de luminosité. Le composant alpha représente la transparence de la couleur.

## hwb()

HWB (abréviation de Hue-Whiteness-Blackness) est une autre méthode de spécification des couleurs, similaire à HSL.

## lab()

La couleur du laboratoire est indépendante de l'appareil et spécifie les mesures physiques de la couleur.

## lch()

La couleur LCH est indépendante de l'appareil et spécifie la couleur à l'aide de

La couleur est indépendante de l'appareil et spécifie la couleur à l'aide de coordonnées polaires pour la chrominance et la teinte.

## [rgb\(\)](#)

Le modèle de couleur RVB définit une couleur donnée en fonction de ses composantes rouge, verte et bleue. Un composant alpha facultatif représente la transparence de la couleur.

## [rgba\(\)](#)

Le modèle de couleur RVB définit une couleur donnée en fonction de ses composantes rouge, verte et bleue. Le composant alpha représente la transparence de la couleur.

# Fonctions d'image

Ces fonctions peuvent être utilisées partout où un [<image>](#) est valide comme valeur d'une propriété.

## [conic-gradient\(\)](#)

Les dégradés coniques font progressivement la transition des couleurs autour d'un cercle.

## [image\(\)](#)

Définit un [<image>](#) d'une manière similaire à la [url\(\)](#) fonction, mais avec des fonctionnalités supplémentaires, notamment la spécification de la directionnalité de l'image, la spécification d'images de secours lorsque l'image préférée n'est pas prise en charge

## [image-set\(\)](#)

Une méthode permettant au navigateur de choisir l'image CSS la plus appropriée dans un ensemble donné, principalement pour les écrans à haute densité de pixels.

## [linear-gradient\(\)](#)

Les dégradés linéaires font progressivement la transition des couleurs le long d'une ligne imaginaire.

## [radial-gradient\(\)](#)

Les dégradés radiaux font progressivement la transition des couleurs à partir d'un point central (origine).

## [repeating-linear-gradient\(\)](#)

Est similaire à `linear-gradient()` et prend les mêmes arguments, mais il répète les arrêts de couleur à l'infini dans toutes les directions de manière à couvrir tout son contenant.

### [repeating-radial-gradient\(\)](#)

Est similaire à `radial-gradient()` et prend les mêmes arguments, mais il répète les arrêts de couleur à l'infini dans toutes les directions de manière à couvrir tout son contenant.

### [repeat-conic-gradient\(\)](#)

Est similaire à `conic-gradient()` et prend les mêmes arguments, mais il répète les arrêts de couleur à l'infini dans toutes les directions de manière à couvrir tout son contenant.

### [cross-fade\(\)](#)

Peut être utilisé pour mélanger deux ou plusieurs images à une transparence définie.

### [element\(\)](#)

Définit une `<image>` valeur générée à partir d'un élément HTML arbitraire.

### [paint\(\)](#)

Définit une `<image>` valeur générée avec un PaintWorklet.

## Fonctions compteur

Les fonctions de compteur sont généralement utilisées avec la [content](#) propriété, bien qu'en théorie elles puissent être utilisées partout où a [<string>](#) est pris en charge.

### [counter\(\)](#)

Renvoie une chaîne représentant la valeur actuelle du compteur nommé, s'il y en a un.

### [counters\(\)](#)

Active les compteurs imbriqués, renvoyant une chaîne concaténée représentant les valeurs actuelles des compteurs nommés, s'il y en a.

### [symbols\(\)](#)

Vous permet de définir des styles de compteur en ligne, directement en tant que valeur d'une propriété.

## Fonctions de police

La [font-variant-alternates](#) propriété contrôle l'utilisation de glyphes alternatifs, les fonctions suivantes sont des valeurs pour cette propriété.

### [stylistic\(\)](#)

Cette fonction permet des alternatives stylistiques pour des caractères individuels. Le



paramètre est un nom spécifique à la police associé à un nombre. Il correspond à la valeur OpenType `salt`, comme `salt 2`.

### [stylesheet\(\)](#)

Cette fonction permet des alternatives stylistiques pour les ensembles de caractères. Le paramètre est un nom spécifique à la police associé à un nombre. Il correspond à la valeur OpenType `ssXY`, comme `ss02`.

### [character-variant\(\)](#)

Cette fonction permet des alternatives stylistiques spécifiques pour les caractères. Il est similaire à `stylesheet()`, mais ne crée pas de glyphes cohérents pour un ensemble de caractères ; les personnages individuels auront des styles indépendants et pas nécessairement cohérents. Le paramètre est un nom spécifique à la police associé à un nombre. Il correspond à la valeur OpenType `cvXY`, comme `cv02`.

### [swash\(\)](#)

Cette fonction permet [orné](#) glyphes. Le paramètre est un nom spécifique à la police associé à un nombre. Il correspond aux valeurs OpenType `swsh` et `cswh`, comme `swsh 2` et `cswh 2`.

### [ornaments\(\)](#)

Cette fonction permet des ornements, comme [fleurons](#) et d'autres glyphes dingbat. Le paramètre est un nom spécifique à la police associé à un nombre. Il correspond à la valeur OpenType `ornm`, comme `ornm 2`.

### [annotation\(\)](#)

Cette fonction active les annotations, comme les chiffres encerclés ou les caractères inversés. Le paramètre est un nom spécifique à la police associé à un nombre. Il correspond à la valeur OpenType `nalt`, comme `nalt 2`.

## Fonctions de forme

Les fonctions suivantes peuvent être utilisées comme valeurs pour le `<basic-shape>` type de données, qui est utilisé dans les propriétés [clip-path](#), [offset-path](#) et [shape-outside](#).

### [circle\(\)](#)

Définit un cercle.

### [ellipse\(\)](#)

Définit une ellipse.

### [inset\(\)](#)

Définit un rectangle d'encart.

### [polygon\(\)](#)

Définit un polygone.

### [path\(\)](#)

Accepte une chaîne de chemin SVG pour permettre le dessin d'une forme.

## Fonctions de référence

Les fonctions suivantes sont utilisées comme valeur de propriétés pour référencer une valeur définie ailleurs.

### [attr\(\)](#)

Permet de récupérer la valeur d'un attribut de l'élément sélectionné et de l'utiliser dans la feuille de style.

### [env\(\)](#)

Utilisé pour insérer la valeur d'une variable d'environnement définie par l'agent utilisateur.

### [url\(\)](#)

Utilisé pour inclure un fichier.

### [var\(\)](#)

Utilisé pour insérer une valeur d'une propriété personnalisée au lieu d'une partie d'une valeur d'une autre propriété.

## Fonctions de grille CSS

Les fonctions suivantes sont utilisées pour définir une grille CSS.

### [fit-content\(\)](#)

Fixe une taille donnée à une taille disponible selon la formule  $\min(\text{maximum size}, \max(\text{minimum size}, \text{argument}))$ .

### [minmax\(\)](#)

Définit une plage de taille supérieure ou égale à *min* et inférieure ou égale à *max*.

### [repeat\(\)](#)

Représente un fragment répété de la liste des pistes, autorisant un grand nombre de colonnes ou de lignes présentant un modèle récurrent.

## Caractéristiques

## Caractéristiques

spécification	Statut	Commenter
---------------	--------	-----------

<a href="#">Module Valeurs et Unités CSS Niveau 4</a>	Brouillon de l'éditeur	Ajoute <code>toggle()</code> , <code>attr()</code> , <code>calc()</code> , <code>min()</code> , <code>max()</code> , <code>clamp()</code> , <code>round()</code> , <code>mod()</code> , <code>rem()</code> , <code>sin()</code> , <code>cos()</code> , <code>tan()</code> , <code>asin()</code> , <code>acos()</code> , <code>atan()</code> , <code>atan2()</code> , <code>pow()</code> , <code>sqrt()</code> , <code>hypot()</code> , <code>log()</code> , <code>exp()</code> , <code>abs()</code> et <code>sign()</code> la notation fonctionnelle.
<a href="#">Module Valeurs et Unités CSS Niveau 3</a>	Recommandation du candidat	Ajoute <code>calc()</code> une notation fonctionnelle.
<a href="#">Module de couleur CSS niveau 4</a>	Document de travail	Ajoute des syntaxes sans virgule pour les notations <code>rgb()</code> , <code>rgba()</code> , <code>hsl()</code> et <code>hsla()</code> fonctionnelles. Autorise les valeurs alpha dans <code>rgb()</code> et <code>hsl()</code> , transformant <code>rgba()</code> et <code>hsla()</code> en alias (obsolètes) pour eux. Ajoute les fonctions <code>hwb()</code> , <code>device-cmyk()</code> et <code>color()</code> .
<a href="#">Module de couleur CSS niveau 3</a>	Recommandation	Ajoute <code>rgba()</code> , <code>hsl()</code> , <code>hsla()</code> notation fonctionnelle.
<a href="#">Module Images CSS Niveau 4</a>	Document de travail	Ajoute <code>element()</code> , <code>image()</code> et la notation fonctionnelle <code>image-set().conic-gradient()</code>

## Voir également

- [Valeurs et unités CSS](#)
- [Introduction à CSS : valeurs et unités](#)

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

