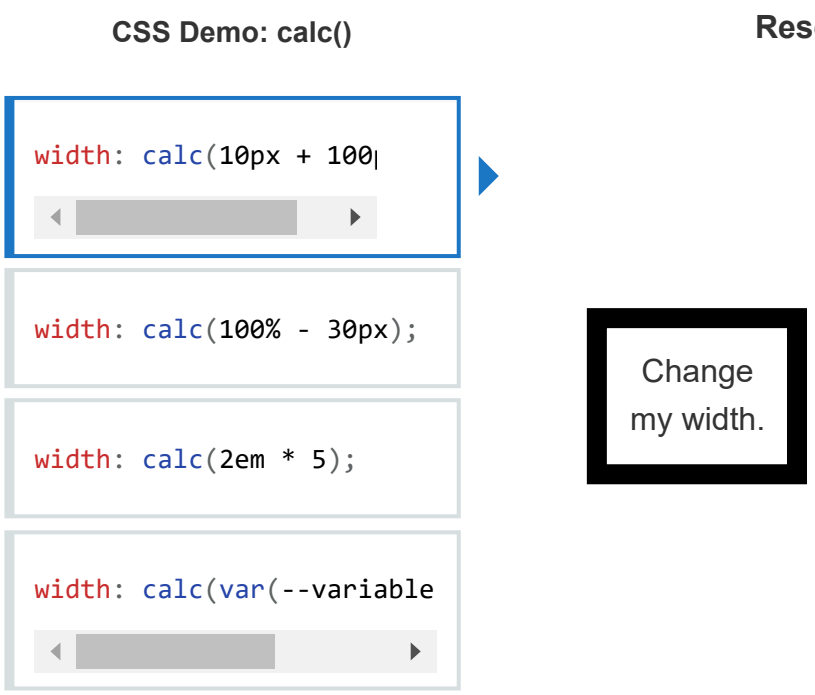
calcul()

La fonction CSS vous permet d'effectuer des calculs lors de la spécification des valeurs des propriétés CSS. Il peut être utilisé partout où un , , , , , ou est autorisé. **`calc()`**
`<length>` `<frequency>` `<angle>` `<time>` `<percentage>` `<number>` `<integer>`



Syntaxe

```
/* property: calc(expression) */
width: calc(100% - 80px);
```

La `calc()` fonction prend une seule expression comme paramètre, avec le résultat de l'expression utilisé comme valeur. L'expression peut être n'importe quelle expression simple combinant les opérateurs suivants, en utilisant les règles standard de priorité des opérateurs :

+
Une addition.

-
Soustraction.

*****

Multiplication. Au moins un des arguments doit être un `<number>`.

`/`
Division. Le côté droit doit être un `<number>`.

Les opérandes de l'expression peuvent être n'importe quelle `<length>` valeur de syntaxe. Vous pouvez utiliser différentes unités pour chaque valeur de votre expression, si vous le souhaitez. Vous pouvez également utiliser des parenthèses pour établir l'ordre de calcul si nécessaire.

Remarques

- Les opérateurs `+` et **doivent être entourés d' espaces** . Par exemple, sera analysé comme un pourcentage suivi d'une longueur négative - une expression invalide - tandis que est un pourcentage suivi d'un opérateur de soustraction et d'une longueur. De même, est traité comme une longueur suivie d'un opérateur d'addition et d'un pourcentage négatif. - `calc(50% -8px)` `calc(50% - 8px)` `calc(8px + -50%)`
- Les opérateurs `*` et `/` ne nécessitent pas d'espace, mais leur ajout pour des raisons de cohérence est à la fois autorisé et recommandé.
- La division par zéro entraîne une erreur générée par l'analyseur HTML.
- Les expressions mathématiques impliquant des pourcentages pour les largeurs et les hauteurs des colonnes de tableau, des groupes de colonnes de tableau, des lignes de tableau, des groupes de lignes de tableau et des cellules de tableau dans les tableaux à mise en page automatique et fixe *peuvent* être traitées comme si elles `auto` avaient été spécifiées.
- Il est permis d'imbriquer des `calc()` fonctions, auquel cas les fonctions internes sont traitées comme de simples parenthèses.

Syntaxe formelle

```
calc( <calc-sum> )

where
<calc-sum> = <calc-product> [ [ '+' | '-' ] <calc-product> ]*

where
<calc-product> = <calc-value> [ '*' <calc-value> | '/' <number> ]*

where
<calc-value> = <number> | <dimension> | <percentage> | ( <calc-sum> )
```

Problèmes d'accessibilité

Lorsque `calc()` est utilisé pour contrôler la taille du texte, assurez-vous que l'une des valeurs inclut une unité de longueur relative , par exemple :

```
h1 {
  font-size: calc(1.5rem + 3vw);
}
```

Cela garantit que la taille du texte sera mise à l'échelle si la page est agrandie.

- MDN Comprendre les WCAG, explications de la directive 1.4
- Comprendre le critère de succès 1.4.4 | W3C Comprendre les WCAG 2.0

Utilisation avec des entiers

Lorsque **`calc()`** est utilisé là où an `<integer>` est attendu, la valeur sera arrondie à l'entier le plus proche. Par example:

```
.modal {
  z-index: calc(3 / 2);
}
```

Cela donnera `.modal` une valeur finale `z-index` de 2.

> **Remarque :** Le navigateur Chrome n'accepte actuellement pas certaines valeurs renvoyées par **`calc()`** lorsqu'un nombre entier est attendu. Cela inclut toute division, même si elle aboutit à un nombre entier. c'est à dire. `z-index: calc(4 / 2);` ne sera pas accepté.

Exemples

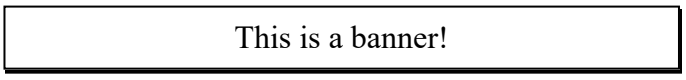
Positionner un objet à l'écran avec une marge

`calc()` permet de positionner facilement un objet avec une marge définie. Dans cet exemple, le CSS crée une bannière qui s'étend sur toute la fenêtre, avec un espace de 40 pixels entre les deux côtés de la bannière et les bords de la fenêtre :

```
.banner {
  position: absolute;
  left: 40px;
  width: calc(100% - 80px);
  border: solid black 1px;
  box-shadow: 1px 2px;
  background-color: yellow;
  padding: 6px;
  text-align: center;
  box-sizing: border-box;
```

```
  box-sizing: border-box;
}
```

```
<div class="banner">This is a banner!</div>
```



Dimensionnement automatique des champs de formulaire pour s'adapter à leur conteneur

Un autre cas d'utilisation `calc()` consiste à garantir que les champs de formulaire tiennent dans l'espace disponible, sans dépasser le bord de leur conteneur, tout en conservant une marge appropriée.

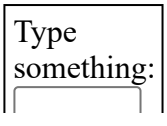
Regardons quelques CSS :

```
input {
  padding: 2px;
  display: block;
  width: calc(100% - 1em);
}

#formbox {
  width: calc(100% / 6);
  border: 1px solid black;
  padding: 4px;
}
```

Ici, le formulaire lui-même est établi pour utiliser 1/6 de la largeur de fenêtre disponible. Ensuite, pour s'assurer que les champs d'entrée conservent une taille appropriée, nous utilisons `calc()` à nouveau pour établir qu'ils doivent être de la largeur de leur conteneur moins 1em. Ensuite, le code HTML suivant utilise ce CSS :

```
<form>
  <div id="formbox">
  <label>Type something:</label>
  <input type="text">
  </div>
</form>
```



Imbriqué `calc()` avec des variables CSS

Vous pouvez également utiliser `calc()` avec des variables CSS . Considérez le code suivant :

```
.foo {
  --widthA: 100px;
  --widthB: calc(var(--widthA) / 2);
  --widthC: calc(var(--widthB) / 2);
  width: var(--widthC);
}
```

Une fois toutes les variables développées, `widthC` la valeur de sera `calc( calc( 100px / 2) / 2)` , puis lorsqu'elle sera affectée à `.foo` la propriété width de , tous les s internes `calc()` (peu importe la profondeur de leur imbrication) seront aplatis en parenthèses, de sorte que la `width` valeur de la propriété sera finalement `calc( ( 100px / 2) / 2)` , c'est-à-dire `25px` . En bref : a `calc()` à l'intérieur de a `calc()` est identique à de simples parenthèses.

Caractéristiques

spécification
Module Valeurs CSS et Unités Niveau 5 (Valeurs CSS 5) # calc-func

Compatibilité du navigateur

Signaler les problèmes avec ces données de compatibilité sur GitHub

`calc()`	
Chrome	26
Edge	12
Firefox	16
Internet Explorer	9
Opera	15
Safari	sept
WebView Android	37

Chrome Android	28
Firefox for Android	16
Opera Android	14
Safari on iOS	sept
Samsung Internet	1.5

Prise en charge des arrêts de couleur dégradés

Chrome	19
Edge	12
Firefox	19
Internet Explorer	9
Opera	15
Safari	6
WebView Android	37
Chrome Android	28
Firefox for Android	19
Opera Android	15
Safari on iOS	6
Samsung Internet	1.5

`calc()` Prise en charge imbriquée

Chrome	51
Edge	16
Firefox	48
Internet Explorer	**Non**
Opera	38
Safari	11
WebView Android	51
Chrome Android	51
Firefox for Android	48
Opera Android	41

Safari on iOS	11
Samsung Internet	5.0

`<number>` soutien de la valeur	
Chrome	31
Edge	12
Firefox	48
Internet Explorer	9
Opera	18
Safari	6
WebView Android	4.4.3
Chrome Android	31
Firefox for Android	48
Opera Android	18
Safari on iOS	6
Samsung Internet	2.0

☐ 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

- Firefox 4 : CSS3 calc() ☆ Mozilla Hacks – le blog des développeurs Web

Dernière modification : 16 novembre 2021, par les contributeurs MDN