

Cette page a été traduite à partir de l'anglais par la communauté. Vous pouvez également contribuer en rejoignant la communauté francophone sur MDN Web Docs.

display

La propriété **display** définit le type d'affichage utilisée pour le rendu d'un élément ([de bloc ou en ligne](#)) et la disposition utilisée pour ses éléments fils : [grille](#) ou [boîtes flexibles](#).

Le type d'affichage donné par **display** possède deux composantes : le type d'affichage extérieur qui définit comment la boîte participe au [flux](#) et le type d'affichage intérieur qui définit l'organisation des éléments enfants.

Certaines valeurs de **display** sont définies dans des spécifications séparées. Pour plus d'informations, voir la section Spécifications ci-après.

Syntaxe

CSS

```
/* Valeurs de type <display-outside> */
display: block;
display: inline;
display: run-in;

/* Valeurs de type <display-inside> */
display: flow;
display: flow-root;
display: table;
display: flex;
display: grid;
display: ruby;

/* Combinaison de valeurs */
/* <display-outside> et <display-inside> */
display: block flow;
display: inline table;
display: flex run-in;

/* Valeurs de type <display-listitem> */
display: list-item;
display: list-item block;
display: list-item inline;
display: list-item flow;
display: list-item flow-root;
display: list-item block flow;
display: list-item block flow-root;
display: flow list-item block;

/* Valeurs de type <display-internal> */
display: table-row-group;
display: table-header-group;
display: table-footer-group;
display: table-row;
display: table-cell;
display: table-column-group;
display: table-column;
display: table-caption;
```

```
display: ruby-base;
display: ruby-text;
display: ruby-base-container;
display: ruby-text-container;

/* Valeurs de type <display-box> */
display: contents;
display: none;

/* Valeurs de type <display-legacy> */
display: inline-block;
display: inline-table;
display: inline-flex;
display: inline-grid;

/* Valeurs globales */
display: inherit;
display: initial;
display: unset;
```

La propriété `display` est définie à l'aide de mots-clés. Ces valeurs sont rangées selon six catégories.

[`<display-outside>`](#)

Ces mots-clés définissent le type d'affichage extérieur de l'élément. Autrement dit, ils définissent comment l'élément participe au flux.

[`<display-inside>`](#)

Ces mots-clés définissent le type d'affichage intérieur de l'élément qui définit le contexte de formatage qui organisera le contenu de l'élément (si celui-ci n'est pas un élément remplacé).

[`<display-listitem>`](#)

L'élément génère une boîte de bloc pour le contenu et une boîte en ligne séparée pour l'élément de liste.

[`<display-internal>`](#)

Certains modes de dispositions, tels que `table` et `ruby` possèdent une structure interne complexe avec différents rôles possibles pour les éléments descendants. Cette section définit ces rôles internes qui s'appliquent uniquement pour un mode de disposition donné.

[`<display-box>`](#)

Ces valeurs définissent si un élément génère une boîte ou non.

[`<display-legacy>`](#)

CSS 2 utilise une syntaxe avec un seul mot-clé pour la propriété `display` et il faut donc des mots-clés distincts pour les variantes bloc/en ligne d'un même mode de disposition.

Valeurs historiques de `display`

La spécification de niveau 3 permet d'utiliser deux valeurs pour définir la propriété `display` afin de définir explicitement le type d'affichage intérieur et le type d'affichage extérieur. Toutefois, cette syntaxe sur deux valeurs n'est pas encore prise en charge de façon homogène par les navigateurs.

Les méthodes d'affichage historiques permettent d'obtenir les mêmes résultats avec un seul mot-clé. Ces valeurs devraient être privilégiées tant que la prise en charge de la syntaxe sur deux valeurs n'est pas mieux prise en charge. Ainsi, il est possible d'utiliser deux valeur afin de définir un conteneur flexible en ligne :

CSS

```
.container {
  display: inline flex;
```

```
}
```

On pourra obtenir un comportement équivalent en écrivant cela avec une seule valeur.

```
CSS

.container {
  display: inline-flex;
}
```

Définition formelle

Valeur initiale	inline
Applicabilité	tous les éléments
Héritée	non
Valeur calculée	comme la valeur spécifiée, excepté pour les éléments positionnés et flottants, ainsi que pour l'élément racine. Dans les deux cas, la valeur calculée peut être un mot clé différent de celui spécifié.
Type d'animation	Discrete behavior except when animating to or from none is visible for the entire duration

Syntaxe formelle

```
display =
  [ <display-outside> | <display-inside> ]
  <display-listitem>
  <display-internal>
  <display-box>
  <display-legacy>
  <display-outside> | <display-inside> | math ]

<display-outside> =
  block |
  inline |
  run-in

<display-inside> =
  flow |
  flow-root |
  table |
  flex |
  grid |
  ruby

<display-listitem> =
  <display-outside>2 &&
  [ flow | flow-root ]2 &&
  list-item

<display-internal> =
  table-row-group |
  table-header-group |
  table-footer-group |
  table-row |
  table-cell |
  table-column-group |
  table-column |
  table-caption |
  ruby-base |
  ruby-text |
  ruby-base-container |
  ruby-text-container
```

```
<display-box> =  
  contents  .  
  none  
  
<display-legacy> =  
  inline-block  .  
  inline-table  .  
  inline-flex   .  
  inline-grid
```

Exemples

En plus de l'exemple qui suit, vous pouvez voir d'autres exemples sur les sections dédiées aux différents modes de disposition :

- [Grilles CSS](#)
- [Boîtes flexibles \(flexbox\) CSS](#)
- [Apprendre la disposition en CSS](#)

HTML

HTML

Play

```
<p>Texte visible</p>  
<p class="secret">Texte invisible</p>
```

CSS

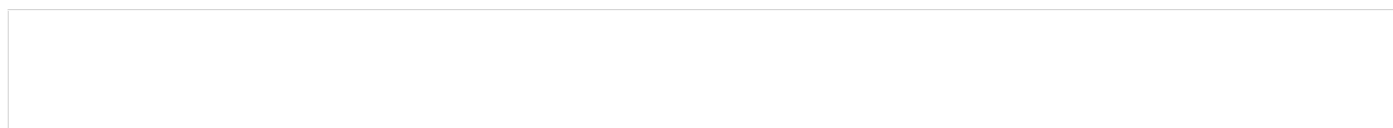
CSS

Play

```
p.secret {  
  display: none;  
}
```

Résultat

Play



Accessibilité

display: none;

Utiliser la propriété `display` avec la valeur `none` sur un élément entraînera son retrait de l'[arbre d'accessibilité](#). Cet élément ainsi que ses descendants ne seront plus annoncés par les lecteurs d'écrans.

Si vous souhaitez masquer un élément visuellement, une alternative plus accessible consiste à utiliser [une combinaison de propriétés](#) afin de le retirer de l'écran mais de le conserver lisible pour les technologies d'assistance.

display: contents;

Tout élément ciblé avec `display: contents` sera retiré de l'arbre d'accessibilité par la plupart des navigateurs. Ainsi, l'élément et ses descendants ne seront plus annoncés par les outils d'assistance tels que les lecteurs d'écran. Ce comportement est incorrect selon [la spécification CSSWG](#).

- [display: contents n'est pas un outil de réinitialisation CSS, par Adrian Roselli, en anglais](#)
- [Obtenir un balisage plus accessible grâce à display: contents, en anglais, par Hidde de Vries](#)

Les tableaux

Modifier la valeur de `display` pour un élément de [tableau](#) afin d'utiliser la valeur `block`, `grid` ou `flex` modifiera sa représentation au sein de l'[arbre d'accessibilité](#). Aussi, le tableau ne sera plus correctement annoncé par les technologies d'assistance.

- [Une rapide note sur l'impact de la propriété CSS `display` sur la sémantique des tableaux — The Paciello Group \(en anglais\)](#).
- [Du contenu masqué avec une meilleure accessibilité - Go Make Things \(en anglais\)](#).
- [Comprendre les règles WCAG 1.3](#)
- [Understanding Success Criterion 1.3.1, W3C Understanding WCAG 2.0 \(en anglais\)](#).


Spécifications

Spécification
CSS Display Module Level 3 # the-display-properties

Compatibilité des navigateurs

[Report problems with this compatibility data on GitHub](#)

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android
<code>display</code>	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome Android 18	Firefox for Android
<code>contents</code>	Chrome 65	Edge 79	Firefox 37	Opera 52	Safari 11.1	Chrome Android 65	Firefox for Android
Specific	Chrome 65	Edge 79	Firefox 59	Opera 52	Safari No	Chrome 65	Firefox for Android

 [mdn web docs](#)

Elements when <code>display: contents</code> is applied to them	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android
<display-outside>	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome Android 18	Firefox for Android
<code>flex</code>	Chrome 29	Edge 12	Firefox 20	Opera 16	Safari 9	Chrome Android 29	Firefox for Android
<code>flow-root</code>	Chrome 58	Edge 79	Firefox 53	Opera 45	Safari 13	Chrome Android 58	Firefox for Android
<code>grid</code>	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 10.1	Chrome Android 57	Firefox for Android
<code>inline-block</code>	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome Android 18	Firefox for Android

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android
inline-flex	Chrome 29	Edge 12	Firefox 20	Opera 16	Safari 9	Chrome 29 Android	Firefox 20 for Android
inline-grid	Chrome 57	Edge 16	Firefox 52	Opera 44	Safari 10.1	Chrome 57 Android	Firefox 52 for Android
inline-table	Chrome 1	Edge 12	Firefox 3	Opera 7	Safari 1	Chrome 18 Android	Firefox 3 for Android
Transitionable when setting transition-behavior: allow-discrete	Chrome 117	Edge 117	Firefox No	Opera 103	Safari No	Chrome 117 Android	Firefox 103 for Android
@keyframe animatable	Chrome 116	Edge 116	Firefox No	Opera 102	Safari No	Chrome 116 Android	Firefox 102 for Android
list-item	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome 18 Android	Firefox 1 for Android
Supported on <legend>	Chrome 71	Edge 79	Firefox 64	Opera 58	Safari No	Chrome 71 Android	Firefox 64 for Android
math	Chrome 109	Edge 109	Firefox No	Opera 95	Safari No	Chrome 109 Android	Firefox 95 for Android
Multi-keyword values	Chrome 115	Edge 115	Firefox 70	Opera 101	Safari 15	Chrome 115 Android	Firefox 70 for Android
none	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome 18 Android	Firefox 1 for Android
ruby	Chrome No	Edge 12–18	Firefox 38	Opera No	Safari No	Chrome No Android	Firefox 38 for Android
ruby-base	Chrome No	Edge 12–18	Firefox 38	Opera No	Safari No	Chrome No Android	Firefox 38 for Android
ruby-base-container	Chrome No	Edge 12–18	Firefox 38	Opera No	Safari No	Chrome No Android	Firefox 38 for Android
ruby-text	Chrome No	Edge 12–18	Firefox 38	Opera No	Safari No	Chrome No Android	Firefox 38 for Android
ruby-text-container	Chrome No	Edge 12–18	Firefox 38	Opera No	Safari No	Chrome No Android	Firefox 38 for Android
table	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome 18 Android	Firefox 1 for Android
table-cell	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome 18 Android	Firefox 1 for Android
table-column	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome 18 Android	Firefox 1 for Android
table-column-group	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome 18 Android	Firefox 1 for Android

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android
table-footer-group	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome Android 18	Firefox for Android 18
table-header-group	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome Android 18	Firefox for Android 18
table-row	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome Android 18	Firefox for Android 18
table-row-group	Chrome 1	Edge 12	Firefox 1	Opera 7	Safari 1	Chrome Android 18	Firefox for Android 18

Tip: you can click/tap on a cell for more information.

- Full support

No support

Experimental. Expect behavior to change in the future.

See implementation notes.
- Requires a vendor prefix or different name for use.

Has more compatibility info.

Voir aussi

- [visibility](#)
- [float](#)
- [position](#)
- [flex](#)
- [grid](#)
- [Explications sur les contextes de formatage](#)
- [Les dispositions de bloc et en ligne dans un flux normal](#)

Help improve MDN

Was this page helpful to you?

Yes

No

[Learn how to contribute.](#)

This page was last modified on 3 août 2023 by [MDN contributors](#).

