

# visibility

**Baseline** Widely available

The `visibility` [CSS](#) property shows or hides an element without changing the layout of a document. The property can also hide rows or columns in a [table](#).

## Try it

CSS Demo: visibility

RESET

`visibility: visible`

`visibility: hidden;`

`visibility: collapse;`

To both hide an element *and remove it from the document layout*, set the [display](#) property to `none` instead of using `visibility`.

## Syntax

CSS

```
/* Keyword values */
visibility: visible;
visibility: hidden;
visibility: collapse;

/* Global values */
visibility: inherit;
visibility: initial;
visibility: revert;
visibility: revert-layer;
visibility: unset;
```

The `visibility` property is specified as one of the keyword values listed below.

### Values

`visible`

The element box is visible.

`hidden`

The element box is invisible (not drawn), but still affects layout as normal. Descendants of the element will be visible if they have `visibility` set to `visible`. The element cannot receive focus (such as when navigating through [tab indexes](#)).

## collapse

The `collapse` keyword has different effects for different elements:

- For [<table>](#) rows, columns, column groups, and row groups, the row(s) or column(s) are hidden and the space they would have occupied is removed (as if `display: none` were applied to the column/row of the table). However, the size of other rows and columns is still calculated as though the cells in the collapsed row(s) or column(s) are present. This value allows for the fast removal of a row or column from a table without forcing the recalculation of widths and heights for the entire table.
- Collapsed flex items and ruby annotations are hidden, and the space they would have occupied is removed.
- For other elements, `collapse` is treated the same as `hidden`.

## Accessibility

Using a `visibility` value of `hidden` on an element will remove it from the [accessibility tree](#). This will cause the element and all its descendant elements to no longer be announced by screen reading technology.

## Interpolation

When animated, visibility values are interpolated between *visible* and *not-visible*. One of the start or ending values must therefore be `visible` or no [interpolation](#) can happen. The value is interpolated as a discrete step, where values of the easing function between 0 and 1 map to `visible` and other values of the easing function (which occur only at the start/end of the transition or as a result of `cubic-bezier()` functions with y values outside of [0, 1]) map to the closer endpoint.

## Notes

- Support for `visibility: collapse` is missing or partially incorrect in some modern browsers. It may not be correctly treated like `visibility: hidden` on elements other than table rows and columns.
- When applied to table rows, if the table contains cells ([<td>](#), [<tr>](#) elements) that span both visible and collapsed rows, the cell may render in unexpected ways. If the spanning cell is defined in a collapsed row, browsers do not render the table cell, as if the cells in subsequent rows were present with `visibility: collapse` applied. When the cell is defined in a visible row and spans a collapsed row, the cell contents are not reflowed, but the presentation of the cell itself varies by browser. Most browsers reduce the block size of the cell by the block size of the hidden row. This means the contents may be larger than the cell in the block-size direction. Depending on the browser, the overflowing contents are either cropped, as if `overflow: hidden` were set, while the content bleeds into the subsequent row in other browsers as if `overflow: visible` were set. In other browsers, the cell is rendered as if the row were not collapsed, with all the other cells in the row hidden as if `visibility: collapse` were set on individual cells rather than the row itself.
- `visibility: collapse` may change the layout of a table if the table has nested tables within the cells that are collapsed, unless `visibility: visible` is specified explicitly on nested tables.

## Formal definition

<a href="#">Initial value</a>	<code>visible</code>
<b>Applies to</b>	all elements
<a href="#">Inherited</a>	yes

[Computed value](#)

as specified

[Animation type](#)a [visibility](#)

## Formal syntax

```
visibility =  
  visible |  
  hidden |  
  collapse
```

## Examples

### Basic example

#### HTML

```
HTML
```

```
<p class="visible">The first paragraph is visible.</p>  
<p class="not-visible">The second paragraph is NOT visible.</p>  
<p class="visible">  
  The third paragraph is visible. Notice the second paragraph is still occupying  
  space.  
</p>
```

[Play](#)

#### CSS

```
CSS
```

```
.visible {  
  visibility: visible;  
}  
  
.not-visible {  
  visibility: hidden;  
}
```

[Play](#)[Play](#)

### Table example

#### HTML

```
HTML
```

```
<table>  
  <tr>  
    <td>1.1</td>  
    <td class="collapse">1.2</td>  
    <td>1.3</td>  
  </tr>  
  <tr class="collapse">
```

[Play](#)

```
<td>2.1</td>
<td>2.2</td>
<td>2.3</td>
</tr>
<tr>
<td>3.1</td>
<td>3.2</td>
<td>3.3</td>
</tr>
</table>
```

CSS

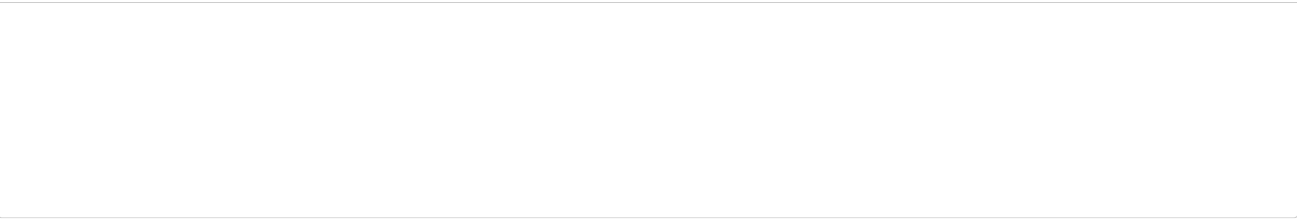
CSSPlay

```
.collapse {
  visibility: collapse;
}

table {
  border: 1px solid red;
}

td {
  border: 1px solid gray;
}
```

Play



Specifications

Specification
<a href="#">CSS Display Module Level 3</a> <a href="#"># visibility</a>

Browser compatibility

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

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
visibility	1	12	1	4	1	18	4	10.1	1	1.0	4.4	1
collapse	1	12	1	4	1.3	18	4	10.1	1	1.0	4.4	1

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
hidden	1	12	1	15	1	18	4	14	1	1.0	4.4	1
visible	1	12	1	15	1	18	4	14	1	1.0	4.4	1

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

Full support      See implementation notes.

See also

- [display](#).

Help improve MDN

Was this page helpful to you?

Yes

No

[Learn how to contribute.](#)

This page was last modified on Dec 19, 2024 by [MDN contributors](#).

