

Description

The *display* property affects the most basic presentation of an element, effectively classing the element as a certain type of element. The rendering of the element may depend heavily on its display type, and certain properties will only work on elements that have specific display values.

Possible Values

- **inline** – This value causes an element to generate an inline-level box; for example, the HTML elements STRONG, CODE, or EM *amongothers*. The element will generate one or more inline boxes when it is displayed.
- **block** – This value causes an element to generate a block-level box; for example the HTML elements P, H1, or PRE *amongothers*. The element will generate a block box when it is displayed.
- **list-item** – This value causes an element to generate both a block box and a list-item inline box. In HTML, the LI element is the only example of such an element.
- **run-in** – Under certain conditions, this value will cause the element to be .inserted. into the beginning of the following element. If an element A is set to display: run-in and is followed by a block-level element B, then A becomes the first inline-level box of B. If the element following A is not block-level, then A becomes a block-level box.
- **compact** – Under certain conditions, this value will cause the element to be placed to one side of the following element.
- **marker** – This value will set generated content to be a marker; thus, it should be used only in conjunction with the :before and :after pseudo- elements when they are set on block-level elements.
- **table** – This value causes an element to generate a block-level table box. This is analogous to the HTML element TABLE.
- **inline-table** – This value causes an element to generate an inline-level table box. While there is no analogue in HTML, it can be envisioned as a traditional HTML table which can appear in the middle of a line of text.
- **table-cell** – This value declares the element to be a table cell. This is analogous to the HTML element TD.
- **table-row** – This value declares the element to be a row of table cells. This is analogous to the HTML element TR.
- **table-row-group** – This value declares the element to be a group of table rows. This is analogous to the HTML element TBODY.
- **table-column** – This value declares the element to be a column of table cells. This is analogous to the HTML element COL.
- **table-column-group** – This value declares the element to be a group of table columns. This is analogous to the HTML element COLGROUP.
- **table-header-group** – This value declares the element to be a group of cells which is always visible at the top of the table, placed before any row or row-groups but after any top-aligned table captions. This is analogous to the HTML element THEAD.
- **table-footer-group** – This value declares the element to be a group of cells which is always visible at the bottom of the table, placed after any row or row-groups but before any bottom-aligned table captions. This is analogous to the HTML element TFOOT.
- **table-caption** – This value declares the element to be a caption for a table. This is

analogous to the HTML element CAPTION.

- **none** – The element will generate no boxes at all, and thus will neither be displayed nor impact the layout of the document.

Applies to

All the HTML elements.

DOM Syntax

```
object.style.cueAfter = url("music.wav");
```

Example

Here is the example –

```
<p style="display:inline;">
This paragraph will inline with the next paragraph
</p>

<p style="display:inline;">
and will make a single line.
</p>

<br />
<br />

<div style="display:block;">
This paragraph will be separate from the next parahgraph
</div>

<div style="display:block;">
and this is second paragraph.
</div>
```

This will produce following result –

This paragraph will inline with the next paragraph

and will make a single line.

This paragraph will be separate from the next parahgraph

and this is second paragraph.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js