

justify-self

The [CSS](#) `justify-self` property sets the way a box is justified inside its alignment container along the appropriate axis.

Try it

CSS Demo: justify-self RESET

`justify-self: stretch;`

`justify-self: center;`

`justify-self: start;`


`justify-self: end;`

One	Two
Three	

The effect of this property is dependent of the layout mode we are in:

- In block-level layouts, it aligns an item inside its containing block on the inline axis.
- For absolutely-positioned elements, it aligns an item inside its containing block on the inline axis, accounting for the offset values of top, left, bottom, and right.
- In table cell layouts, this property is *ignored*. Read more about [alignment in block, absolute positioned and table layout](#).
- In flexbox layouts, this property is *ignored*. Read more about [alignment in Flexbox](#).
- In grid layouts, it aligns an item inside its grid area on the inline axis. Read more about [alignment in grid layouts](#).

Syntax

CSS 

```
/* Basic keywords */
justify-self: auto;
justify-self: normal;
justify-self: stretch;

/* Positional alignment */
justify-self: center; /* Pack item around the center */
justify-self: start; /* Pack item from the start */
```

```
justify-self: end; /* Pack item from the end */
justify-self: flex-start; /* Equivalent to 'start'. Note that justify-self is ignored in Flexbox layouts. */
justify-self: flex-end; /* Equivalent to 'end'. Note that justify-self is ignored in Flexbox layouts. */
justify-self: self-start;
justify-self: self-end;
justify-self: left; /* Pack item from the left */
justify-self: right; /* Pack item from the right */

/* Baseline alignment */
justify-self: baseline;
justify-self: first baseline;
justify-self: last baseline;

/* Overflow alignment (for positional alignment only) */
justify-self: safe center;
justify-self: unsafe center;

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

This property can take one of three different forms:

- Basic keywords: one of the keyword values `normal`, `auto`, or `stretch`.
- Baseline alignment: the `baseline` keyword, plus optionally one of `first` or `last`.
- Positional alignment:
 - one of: `center`, `start`, `end`, `flex-start`, `flex-end`, `self-start`, `self-end`, `left`, or `right`.
 - Plus optionally `safe` or `unsafe`.

Values

auto

The value used is the value of the `justify-items` property of the parents box, unless the box has no parent, or is absolutely positioned, in these cases, `auto` represents `normal`.

normal

The effect of this keyword is dependent of the layout mode we are in:

- In block-level layouts, the keyword is a synonym of `start`.
- In absolutely-positioned layouts, the keyword behaves like `start` on *replaced* absolutely-positioned boxes, and as `stretch` on *all other* absolutely-positioned boxes.
- In table cell layouts, this keyword has no meaning as this property is *ignored*.
- In flexbox layouts, this keyword has no meaning as this property is *ignored*.
- In grid layouts, this keyword leads to a behavior similar to the one of `stretch`, except for boxes with an aspect ratio or an intrinsic sizes where it behaves like `start`.

start

The item is packed flush to each other toward the start edge of the alignment container in the appropriate axis.

end

The item is packed flush to each other toward the end edge of the alignment container in the appropriate axis.

flex-start

For items that are not children of a flex container, this value is treated like `start`.

flex-end

For items that are not children of a flex container, this value is treated like `end`.

self-start

The item is packed flush to the edge of the alignment container of the start side of the item, in the appropriate axis.

self-end

The item is packed flush to the edge of the alignment container of the end side of the item, in the appropriate axis.

center

The items are packed flush to each other toward the center of the alignment container.

left

The items are packed flush to each other toward the left edge of the alignment container. If the property's axis is not parallel with the inline axis, this value behaves like `start`.

right

The items are packed flush to each other toward the right edge of the alignment container in the appropriate axis. If the property's axis is not parallel with the inline axis, this value behaves like `start`.

baseline, first baseline, last baseline

Specifies participation in first- or last-baseline alignment: aligns the alignment baseline of the box's first or last baseline set with the corresponding baseline in the shared first or last baseline set of all the boxes in its baseline-sharing group. The fallback alignment for `first baseline` is `start`, the one for `last baseline` is `end`.

stretch

If the combined size of the items is less than the size of the alignment container, any `auto`-sized items have their size increased equally (not proportionally), while still respecting the constraints imposed by `max-height` / `max-width` (or equivalent functionality), so that the combined size exactly fills the alignment container.

safe

If the size of the item overflows the alignment container, the item is instead aligned as if the alignment mode were `start`.

unsafe

Regardless of the relative sizes of the item and alignment container, the given alignment value is honored.

Formal definition

Initial value	auto
Applies to	block-level boxes, absolutely-positioned boxes, and grid items
Inherited	no
Computed value	as specified
Animation type	discrete

Formal syntax

```
justify-self =
  auto                .|.
  normal              .|.
  stretch            .|.
  <baseline-position> .|.
  <overflow-position>? [. | <self-position> .| left .| right .| .|
  anchor-center

<baseline-position> =
  [. first .| last ]? &&
  baseline

<overflow-position> =
  unsafe .|.
  safe

<self-position> =
  center      .|.
  start       .|.
  end         .|.
  self-start  .|.
  self-end    .|.
  flex-start  .|.
  flex-end
```


Examples

Simple demonstration


In the following example we have a simple 2 x 2 grid layout. Initially the grid container is given a `justify-items` value of `stretch` — the default — which causes the grid items to stretch across the entire width of their cells.

The second, third, and fourth grid items are then given different values of `justify-self`, to show how these override the `justify-items` value. These values cause the grid items to span only as wide as their content width, and align in different positions across their cells.

HTML

```
HTML
</> Play 
<article class="container">
  <span>First child</span>
  <span>Second child</span>
  <span>Third child</span>
  <span>Fourth child</span>
</article>
```

CSS

```
CSS
</> Play 
html {
  font-family: helvetica, arial, sans-serif;
  letter-spacing: 1px;
}

article {
  background-color: red;
  display: grid;
  grid-template-columns: 1fr 1fr;
  grid-auto-rows: 40px;
  grid-gap: 10px;
  margin: 20px;
  width: 300px;
  justify-items: stretch;
```

```
}

span:nth-child(2) {
  justify-self: start;
}

span:nth-child(3) {
  justify-self: center;
}

span:nth-child(4) {
  justify-self: end;
}

article span {
  background-color: black;
  color: white;
  margin: 1px;
  text-align: center;
}

article,
span {
  padding: 10px;
  border-radius: 7px;
}
```

Result

⌕ Play

First child

Second child

Third child

Fourth child

Specifications

Specification
CSS Box Alignment Module Level 3 # justify-self-property

Browser compatibility

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

	Desktop					Android			
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	
justify-self	✓ Chrome 57	✓ Edge 16	✓ Firefox 45	✓ Opera 44	✓ Safari 10.1	✓ Chrome 57 Android	✓ Firefox 45 for Android	✓ Opera 43 Android	✓
Supported in Flex Layout	✓ Chrome 57	✓ Edge 16	✓ Firefox 45	✓ Opera 44	✓ Safari 10.1	✓ Chrome 57 Android	✓ Firefox 45 for Android	✓ Opera 43 Android	✓
Supported in Grid Layout	✓ Chrome 57	✓ Edge 16	✓ Firefox 45	✓ Opera 44	✓ Safari 10.1	✓ Chrome 57 Android	✓ Firefox 45 for Android	✓ Opera 43 Android	✓

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

✓ Full support

See also

- CSS Grid Guide: [Box alignment in CSS Grid layouts](#)
- [CSS Box Alignment](#)
- The `justify-items` property

This page was last modified on Jul 17, 2023 by [MDN contributors](#).