

CSS FLEXBOX (flexible box)

for responsive positioning
of elements for various
browsers & devices etc

`display: flex;`

Placing the CSS property `display: flex;` on an element allows you to use other flex properties to build a responsive page.

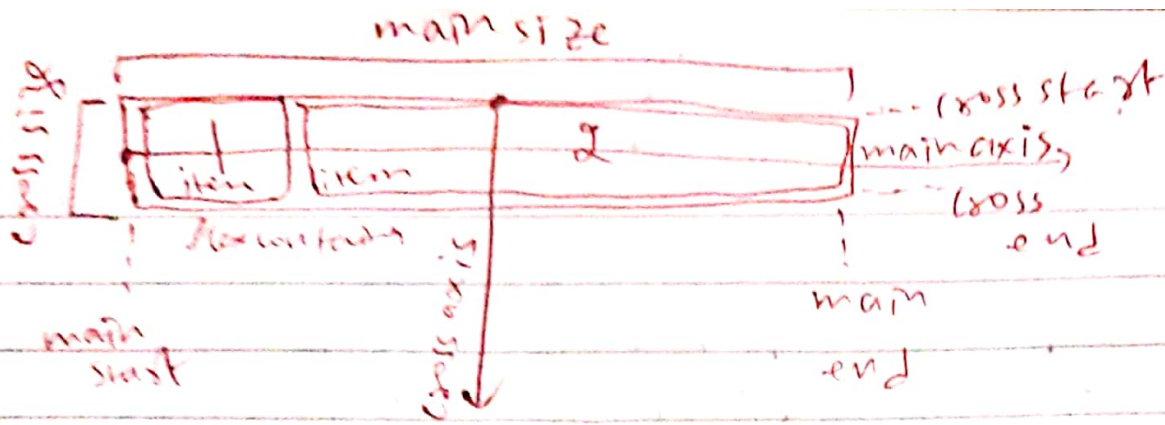
Use the 'flex-direction' property to make a row:

Adding `display: flex;` to an element turns it into a flex container. This makes it possible to align any children of that element into rows or columns. You do this by adding the 'flex-direction' property to the parent item and setting it to row or column. Other options for 'flex-direction' are 'row-reverse' and 'column-reverse'. Default value for flex-direction property is 'row'.

Align elements using the 'justify-content' property:

Sometimes the flex items within a flex container do not fill all the space in the container. It is common to want to tell CSS how to align and space out the flex items a certain way. Fortunately the 'justify-content' property has several options to do this.

=> The direction the flex items are arranged is called main axis.



- justify-content: center; (aligns items to center)
- justify-content: flex-start; (floats to the starting)
- justify-content: flex-end; (floats to the ending)
- justify-content: space-between; (first & last items locked to edges)
- justify-content: space-around; (same as space-between but 1st & last items have half space between the edges)
- justify-content: space-evenly; (evenly spaced)

Align items using 'align-items' property

- 'align-items' is used to align-items along the cross axis
- align-items: flex-start;
- align-items: flex-end;
- align-items: center;
- align-items: stretch; (stretch to fill)
- align-items: baseline; (Baseline is a text concept, think of it as the line that the letters sit on).

Use the 'flex-wrap' property to wrap a row or column

CSS flexbox has a feature to split a flex item into multiple rows (or columns). By default, a flex container will fit all flex items together. For example, a row will all be on one line.

However, using the 'flex-wrap' property tells CSS to wrap items. This means extra items move into a new row or column. The break point of where the wrapping happens depends on the size of the items & of the container.

`flex-wrap: nowrap;` (default, does not wrap items)

`flex-wrap: wrap;` (from left to right wrap for row
from top to bottom wrap for column)

`flex-wrap: wrap-reverse;`

Use the flex-shrink property to shrink items & flex-grow to grow items.

Items shrink when the size of the parent container is smaller than the combined widths of all the flex items within it.

'flex-shrink' property takes numbers as values.

Higher the number, more it will shrink compared to other items. Ex: one item has value 1 and other has 3 then second item will shrink three times as much as other.

Similarly if one ~~item~~ item has a flex-grow value of 1 and other has flex-value of value of 3, the one with value of 3 will grow three times as much as other.

Use the 'flex-basis' property to set the initial size of an item:-

The flex-basis property specifies the initial size of the item before CSS makes adjustments with flex-shrink and flex-grow.

The units used by flex-basis are same as other size properties (pt, em, %, etc). The value auto sizes items based on the content.

Use the flex shorthand property:-

flex: 1 0 10px;

flex-grow flex-shrink flex-basis.

default property settings are flex: 0 1 auto.

Use the 'order' property to rearrange items:-

The order property is used to tell CSS the order of how flex items appear in the flex container. By default, items will appear in the same order they come in the source HTML. The property takes numbers as values, and negative numbers can be used.

Use the align-self property:-

This property allows you to adjust each item's alignment individually, instead of setting them all at once. This is useful since other common adjustment techniques using the CSS properties float, clear and vertical-align do not work on flex items.

'align-self' accepts the same values as 'align-items' and will override any set by the 'align-items' property.