

flex:

- puts things in a row
- causes its children to be as tall as it is

flex-direction:

- row
- column
- row-reverse
- column-reverse

-> Depending on the direction, align-items and justify-content will behave differently.

By default align-items is on the y axis. However, if we change direction to column, it changes to the x-axis, and vice versa.

properties that remove default stretch:

X AXIS:

- align-items: flex-start;
- align-items: flex-end;
- align-items: center;

Y AXIS:

- justify-content: flex-start;
- justify-content: flex-end;
- justify-content: center;

-> If we're going to position something in the center, for example, why does it need to stretch to fit the whole container? Saying it's in the center automatically means we don't want it to touch all the sides of the parent container.

properties that don't remove default stretch:

X AXIS:

- align-items: space-around; - Each child element of the flex box has an invisible margin of equal width. These margins don't overlap, causing double the space between siblings then between sibling and parent element.
- align-items: space-between; - Each child element is as far away from its siblings as possible.
- align-items: space-evenly; - Same as space-around, but the margins do overlap, causing the distance between an element and its sibling to be the same as between an element and its parent.

Y AXIS:

- justify-content: space-around;
- justify-content: space-between;
- justify-content: space-evenly;

flex-order:

-> By giving each of the elements in our container a specific number, we can change the order or position of those elements in our container.

flex-grow, shrink, basis:

-> By changing these properties, we can make our elements have different sizes. Whereas the default behavior of flex is for all children to have the same size.