Difference between align-content and align-item

❖ Align-content:

This property of flex-box aligns flex-lines with respect to each other along the cross-axis.
This property works only when the “flex-wrap:” property is set to wrap
This property has no effect when the number of flex-line is 1.
The align-content property accepts 6 different values:

- **Flex-start: lines packed to the start of the container**
- **Flex-end: lines packed to the end of the container**
- **Center: lines packed to the center of the container**
- **Space-between: lines evenly distributed; the first line is at the start of the container while the last one is at the end**
- **Space-around: lines evenly distributed with equal space between**
- **stretch(Default): lines stretch to take up the remaining space**

❖ Align-items:

This property of flex-box aligns the flex-items within flex-line along the cross-axis
This property works even when the “flex-wrap:” property is not set to wrap
This property does not depend on the number of flex-lines.
The align-items property accepts 5 different values:

- **Flex-start: cross-start margin edge of the items is placed on the cross-start line**
- **Flex-end: cross-end margin edge of the items is placed on the cross-end line**
- **Center: items are centered in the cross-axis**
- **Baseline: items are aligned such as their baselines align**
- **stretch(Default): stretch to fill the container**

Difference between border-box and content-box

❖ ***content-box:***

This is the default value of box-sizing. The dimension of element only includes 'height' and 'width' and does not include 'border' and 'padding' given to element. Padding and Border take space outside the element.

❖ **border-box:**

In this value, not only width and height properties are included but you will find padding and border inside of the box for example .box {width: 200px; border: 10px solid black;} renders a box that is 200px wide.

➢ **Note :- when using box-sizing : content-box; the content size remain same while border-box size grows as padding and border grow. but**

when using box-sizing: border-box; , the size of border-box remains same while size of content decreases as padding and border grow.

Rem and em

- ➔ em units in CSS are a relative unit of measurement used to size elements on a web page, mostly font size. Because it's relative to its parent element, 1 em is equal to the font-size set in the parent element.

This means if you set the font-size in a parent div to 20px and set the font-size of the child div to 2 em, the font-size in the child div will equal 40px.

The em unit is useful because it allows you to adjust the size of elements on a page based on the font-size of a previously stated element, which helps create a consistent visual hierarchy. This can be useful for creating accessible websites that are easy to read for users with visual impairments.

- ➔ rem is another unit of measuring length in CSS, which stands for "root em". Since we know that an em is equal to the point size of the current font, we can deduce that the "root em" refers to the font size of the root element, which is usually the <html> element.

rem inherits its size from a parent element, but the parent element that rem looks at is not the div or section above it, but the very first element that surrounds it, which is the html element. Let's make an example using the previous code. Same html code, with just an extra div.