**Display**

* There are multiple types of display that is mentioned below:

1. Block
2. Inline
3. Inline-block
4. Flex
5. Grid
6. None
7. Table
8. List-item

|  |  |  |
| --- | --- | --- |
| Inline | Displays an element as an inline element (like <span>). Any height and width properties will have no effect | [Demo ❯](https://www.w3schools.com/cssref/playdemo.php?filename=playcss_display&preval=inline) |
| Block | Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width |  |
| Inline-block | It is used to enable the height and width when we want to change the content height or width |  |

|  |  |
| --- | --- |
| None | The element is completely removed |
| list-item | Let the element behave like a <li> element |

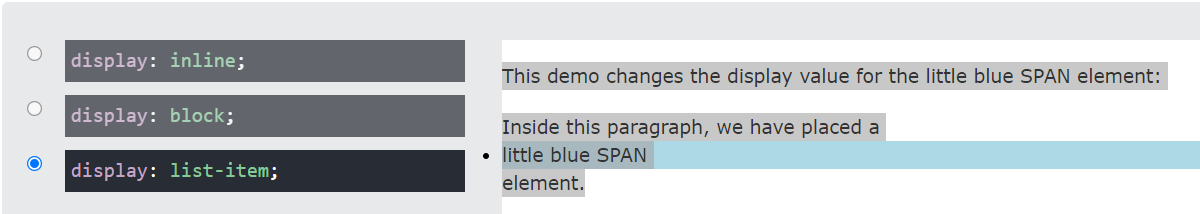
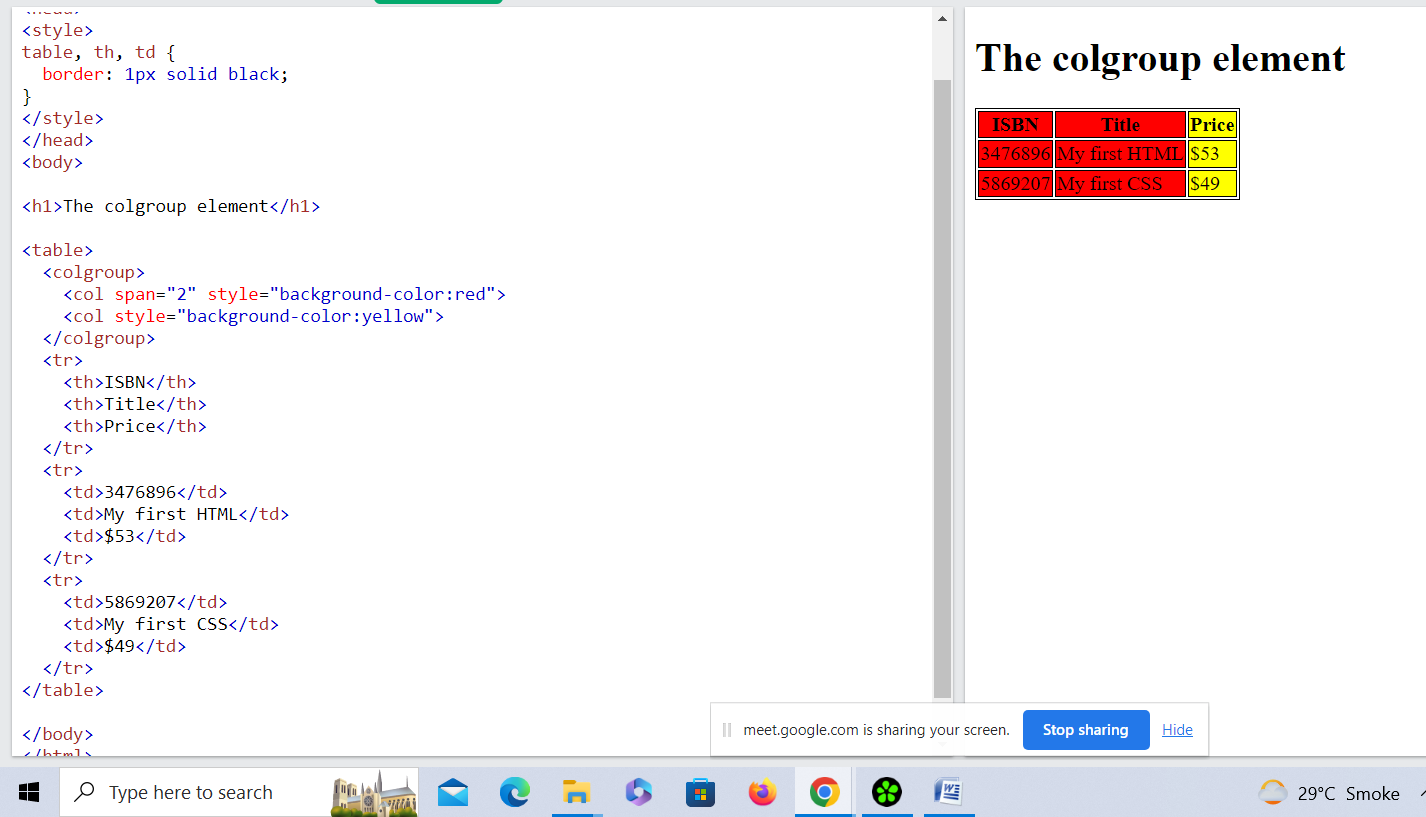


Table Display

1. Colgroup table



|  |
| --- |
|  |
| Table | Let the element behave like a <table> element |  |
| table-caption | Let the element behave like a <caption> element |  |
| table-column-group | Let the element behave like a <colgroup> element |  |
| table-header-group | Let the element behave like a <thead> element |  |
| table-footer-group | Let the element behave like a <tfoot> element |  |
| table-row-group | Let the element behave like a <tbody> element |  |
| table-cell | Let the element behave like a <td> element |  |
| table-column | Let the element behave like a <col> element |  |
| table-row | Let the element behave like a <tr> element |  |

**Display Flex**

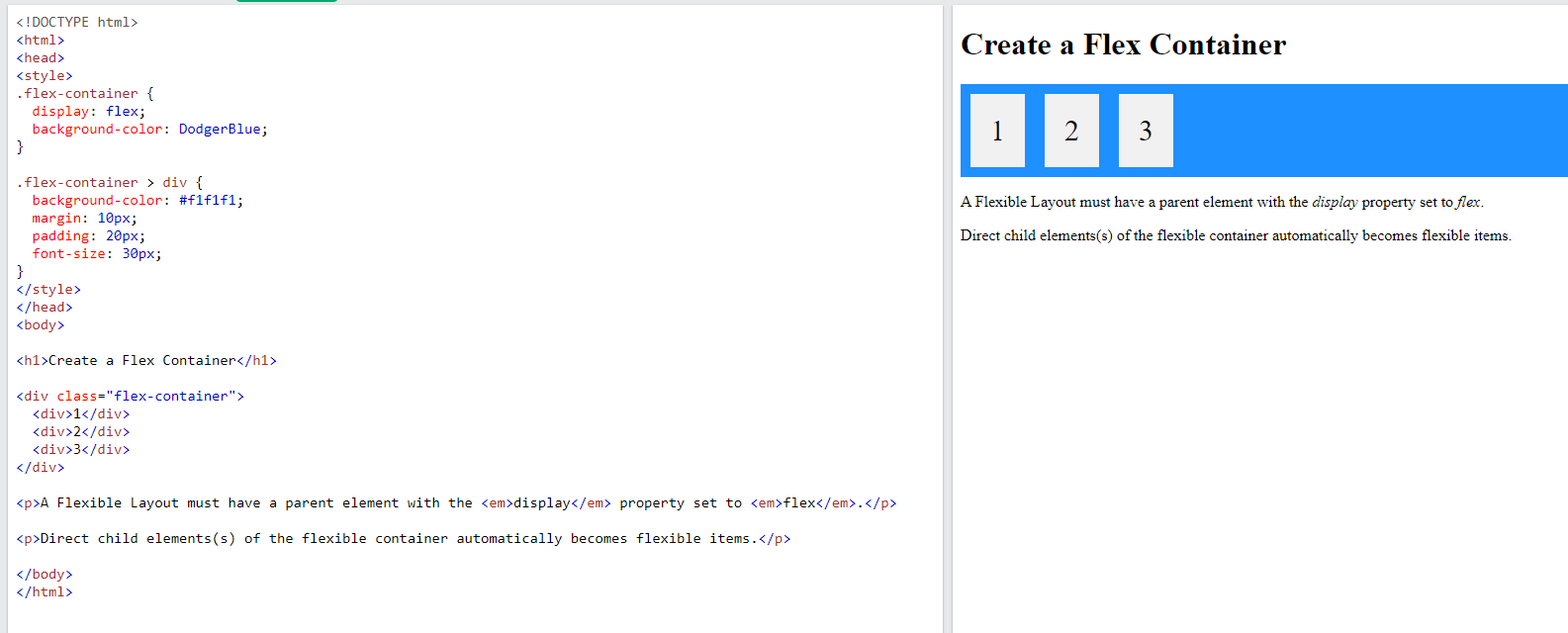
**flex: 0 auto;**

This is the same as flex: initial; and the shorthand for the default value: flex: 0 1 auto. It sizes the item based on its width/height properties (or its content if not set).

**flex: auto;**

This is equivalent to flex: 1 1 auto. Beware, this is not the default value. It sizes the item based on its width/height properties, but makes it fully flexible so that they absorb any extra space along the main axis.

Display Flex Ex



**Display Flex**:-

* Ordering & Orientation
* flex-direction
* flex-wrap
* flex-flow
* order
* justify-content
* align-items, align-self
* align-content
* Flexibility
* flex-grow
* flex-shrink
* flex-basis
* Flex values

**Display flex-direction:**

            display: flex;

            flex-direction: column;

            flex-direction: column-reverse;

            flex-direction: row;

            flex-direction: row-reverse;

Display flex-wrap

1. **Flex Without Wrap**

* When there are many contents without wrap there is not go to new line when we inspect it.

1. **Flex Wrap**

* When there are many contents with wrap there is go to new line when we inspect it.

1. Flex Wrap Reverse

* When there are many contents with wrap there and we want to start displaying content to the end then we use flex wrap reverse.

Flex wrap Example:-

 .no-wrap div {

            width: 200px;

        }

        .flex-wrap {

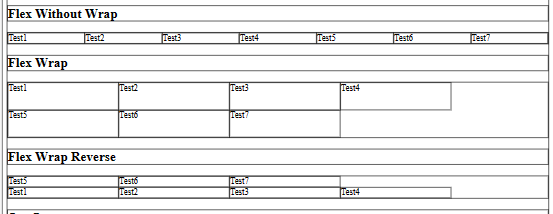
            flex-wrap: wrap;

        }

        .flex-wrap-reverse {

            flex-wrap: wrap-reverse;

        }



1. flex-flow

  .flex-flow-column {

            flex-flow: column;

        }

        .flex-flow-column-reverse{

            flex-flow: column-reverse;

        }

        .flex-flow-row{

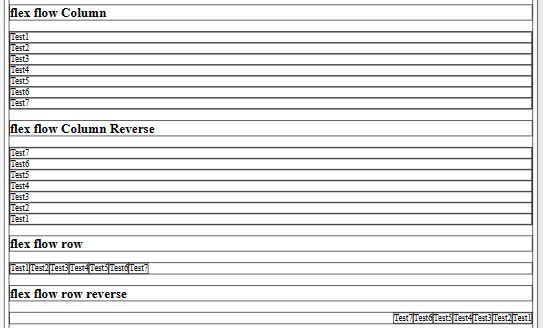
            flex-flow: row;

        }

        .flex-flow-row-reverse{

            flex-flow: row-reverse;

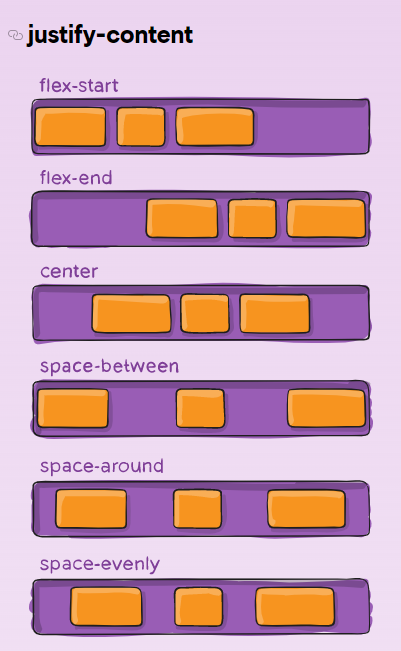
}



1. Order

* The order property specifies the order of a flexible item relative to the rest of the flexible items inside the same container.

1. Justify – content



1. Align items , Align content , Align self

* Align items property is used when we don’t use wrap and when we want to move the content to top to bottom (vertical alignment) then we use align item.
* Align content property is used when we have more than one row and we want to use wrap and then we want to move the content to top to bottom (vertical alignment) then we use align content.
* Align self property is used when we give the css to particular one content then we use align self.

1. Flex grow:-

* The flexGrow property specifies how much the item will grow relative to the rest of the flexible items inside the same container.
* When we use media query when flex grow apply the content is big.



Flex-basis and flex shrink:-

* Flex basis and flex shrink are always together we use.



* When we use media query when flex shrink apply the content is small.