

Utilities

1. What is a lead?

Lead adds some emphasis to a paragraph. The .lead class is used to achieve this and it makes the font larger, taller, and lighter in weight.

```
<p class= "lead" > Paragraph </p>
```

Typography

Use the .lead class to make a paragraph “stand out”:

This paragraph stands out.

This is a regular paragraph.



2. In Bootstrap 4, what is flexbox?

Flexbox is a layout module for flexible boxes. Without using float or positioning, you can quickly create a flexible layout design with flexbox.

3. Discuss Bootstrap table and various classes that can change the appearance of the table.

- A basic Bootstrap 4 table features horizontal divisions and light padding.
- The .table class gives a table some basic styling.
- The .table-striped class gives zebra stripes to the table.
- The .table-bordered class adds borders to the table and cells on all sides.
- On table rows, the .table-hover class adds a hover effect (grey background color).

Here is how the zebra-striped table looks like:

First Name	Last Name
Albert	Halley
Edwin	Powell
Marie	Curie



4. Discuss the properties of flexbox

The properties of flexbox are:

- **Enable flex behaviors:**

To make a flexible container and convert direct children elements into flex items, use the "display" utility.

Additional flex characteristics can be used to further customize flex containers and items.

Default flexbox container:

```
<div class="d-flex p-2">flexbox container</div>
```

You can also create an inline flexbox container. Responsive variations also exist.

- **Direction:**

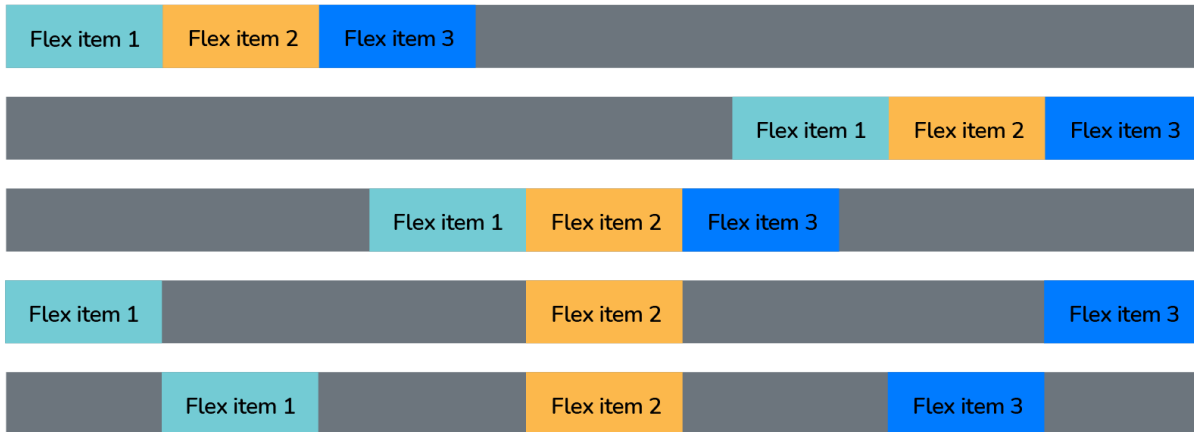
You can use direction utilities to change the direction of a flex item in a flex container. It is not essential to create a horizontal class because it is set to row by default.

To show the flex items horizontally, use .flex-row (side by side). To right-align the horizontal direction, use .flex-row-reverse.

Use .flex-column to display flex items vertically (on top of each other), or .flex-column-reverse to display them in the other direction.

- **Justify content:**

To adjust the alignment of flex elements, use the .justify-content-* classes. Start (default), end, center, between, and around are all valid classes.



- **Align items:**

You can change the vertical alignment of single rows of flex items with the `.align-items-*` classes. `.align-items-start`, `.align-items-end`, `.align-items-center`, `.align-items-baseline`, and `.align-items-stretch` (default) are valid classes.

- **Align self:**

You can change the vertical alignment of a specific flex item with the `.align-self-*` classes. `.align-self-start`, `.align-self-end`, `.align-self-center`, `.align-self-baseline`, and `.align-self-stretch` (default) are valid classes.

- **Auto margins:**

You can easily add auto margins to flex items with `.mr-auto` which pushes items to the right, or by using `.ml-auto` which pushes items to the left.

- **Grow:**

You can use `.flex-grow-1` on a flex item to take up the unused space. In the example below, the second flex item uses all the available space since we have used the `.flex-grow-1` class with the second flex item.



You can use `.flex-shrink-1` on a flex item to make it shrink if required.

- **Fill:**

You can use `.flex-fill` on flex items to make them into equal widths.

- **Wrap:**

With `.flex-nowrap` (default), `.flex-wrap`, or `.flex-wrap-reverse`, you can control how flex items wrap in a flex container.

flex-wrap



flex-wrap-reverse



flex-nowrap



- **Order:**

With the `.order` classes, you can change the visual order of a given flex item(s). The number of valid classes ranges from 0 to 12, with the lowest number having the highest importance (order-1 is shown before order-2, etc..).

- **Align content:**

You can change the vertical alignment of gathered flex items with the `.align-content-*` classes. `.align-content-start` (default), `.align-content-end`, `.align-content-center`, `.align-content-baseline`, and `.align-content-stretch` are valid classes