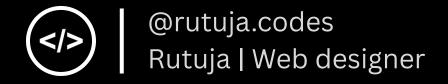
Day 08

CSS Flexbox

CSS Learning series







What is Flexbox?

Flexbox is short for the Flexible Box Layout module.

Flexbox is a layout method for arranging items in rows or columns.

Flexbox makes it easier to design a flexible responsive layout structure, without using float or positioning.

It is particularly useful for:

- Building responsive layouts
- Centering content (vertically and horizontally)
- Controlling spacing and alignment of items

```
.container {
  display: flex;
}
```



Key Terminology

- Flex Container: The parent element with display: flex
- Flex Items: The direct children of the flex container
- Main Axis: The primary direction in which flex items are placed (default: row)
- Cross Axis: The perpendicular direction to the main axis



Flex Container Properties

The Flex Container is the parent element where you apply display: flex. It defines the flex context for its direct children (the flex items), allowing them to be arranged along the main axis and cross axis.

1. display: flex

- This activates the Flexbox layout model for the container.
- It turns the container into a flex container and makes its direct children flex items.

```
.container {
  display: flex;
}
```



2. flex-direction

 Defines the direction in which the flex items are placed in the flex container.

- row (default): Items are placed horizontally from left to right.
- row-reverse: Items are placed horizontally, but in reverse order.
- column: Items are placed vertically from top to bottom.
- column-reverse: Items are placed vertically, but in reverse order.



3. flex-wrap

• Controls whether the flex items should wrap onto multiple lines when there is not enough space.

- nowrap (default): Items will stay on a single line, overflowing the container if necessary.
- wrap: Items will wrap onto the next line if needed.
- wrap-reverse: Items will wrap onto the next line, but in reverse order.



4. justify-content

 Aligns the flex items along the main axis (horizontal by default).

Values:

- flex-start (default): Aligns items to the start of the flex container.
- flex-end: Aligns items to the end of the flex container.
- center: Aligns items to the center of the flex container.
- space-between: Distributes items with equal space between them, no space at the ends.
- space-around: Distributes items with equal space around them.
- space-evenly: Distributes items with equal space between and around them.

@rutuja.codes Rutuja | Web designer



5. align-items

 Aligns the flex items along the cross axis (vertical by default).

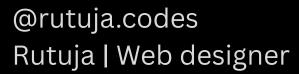
- stretch (default): Stretches items to fill the container's height.
- flex-start: Aligns items at the start of the container.
- flex-end: Aligns items at the end of the container.
- center: Aligns items to the center of the container.
- baseline: Aligns items based on their text baseline.



6. align-content

 Aligns rows of items when the container has multiple lines of flex items (works only when flexwrap is enabled).

- stretch (default): Stretches rows to fill the container's height.
- flex-start: Aligns rows to the start of the container.
- flex-end: Aligns rows to the end of the container.
- center: Aligns rows to the center of the container.
- space-between: Distributes rows with equal space between them.
- space-around: Distributes rows with equal space around them.
- space-evenly: Distributes rows with equal space between and around them.





Flex Item Properties

1. flex-grow

- Determines how much a flex item grows relative to other flex items inside the flex container when extra space is available.
- Default Value: 0 (items don't grow by default).
- Values: Any positive number (integer or float).

```
• • • • . item {
    flex-grow: 1;
}
```



2. flex-shrink

- Defines how much a flex item will shrink relative to other items when there is not enough space in the container.
- Default Value: 1 (items shrink if needed).
- Values: Any positive number (usually integers).
- flex-shrink: 1 means the item will shrink if necessary to fit the container.
- flex-shrink: 0 means the item won't shrink even if the container is too small.

```
.item {
  flex-shrink: 0;
}
```



3. flex-basis

- Defines the initial size of the flex item before any free space is distributed based on flex-grow or flex-shrink.
- Default Value: auto (the item's size based on its content).
- Values: Length values (e.g., px, %, em) or auto.
- If flex-basis: 200px, the item will start with a 200px size before Flexbox applies flex-grow or flex-shrink.
- flex-basis: auto means the item will take its natural size.



4. flex (Shorthand for flex-grow, flex-shrink, and flex-basis)

- A shorthand property to set flex-grow, flex-shrink, and flex-basis in a single line.
- Default Value: 0 1 auto (flex-grow: 0, flex-shrink: 1, flex-basis: auto).
- Values: flex-grow flex-shrink flex-basis (e.g., 1 0 100px).

```
.item {
  flex: 1 1 200px;
}
```



5. align-self

- Allows individual flex items to override the alignitems property of the flex container.
- Default Value: auto (inherits from align-items of the container).
- Values: auto, flex-start, flex-end, center, baseline, stretch.

```
• • • .
.item {
   flex: 1 1 200px;
}
```

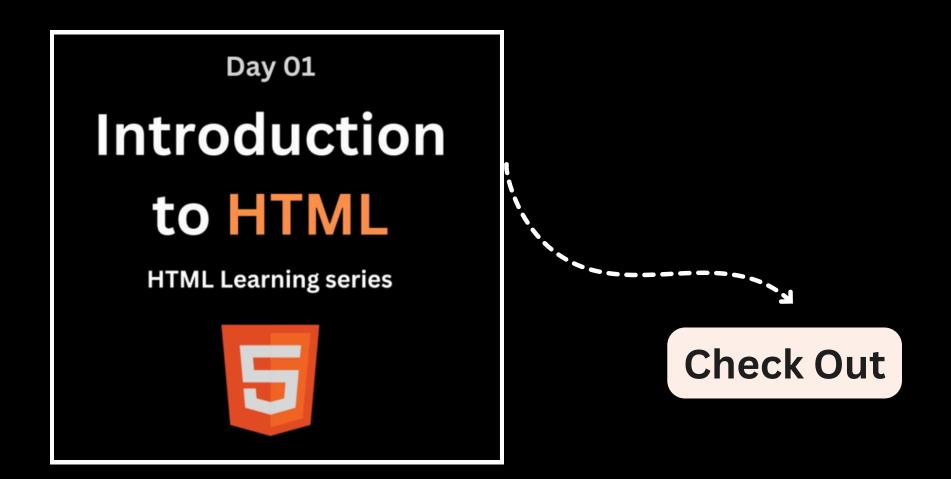


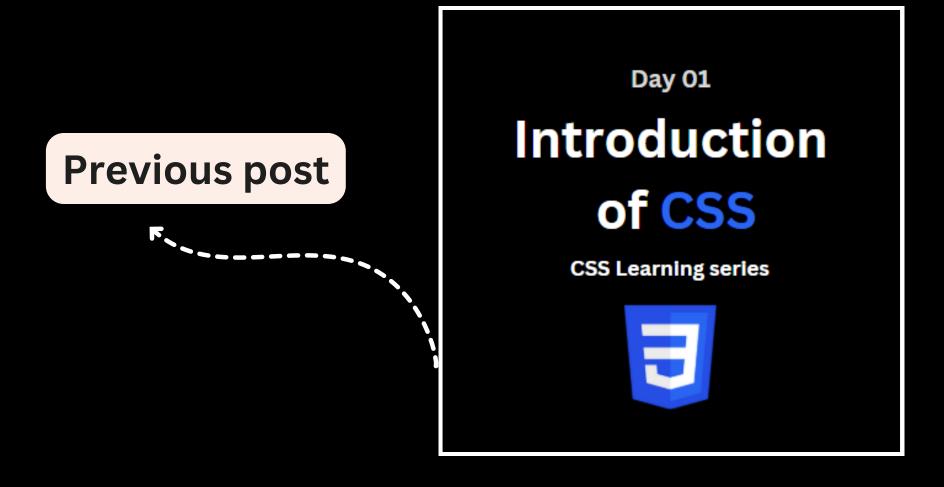
6. order

- Controls the visual order of flex items inside the container, without changing their order in the HTML structure.
- Default Value: 0 (no reordering).
- Values: Any integer (positive or negative).

```
• • • .
.item {
   order: 1;
}
```







@rutuja.codes Rutuja | Web designer



Thank you for learning with me!

- Drop a like
- **Share it with friends**
- Leave a Comment
- Save this post

