**1.FlexBox**

**-> Flex Direction**

- Utilities for controlling the direction of flex items.

**<-> Classes:**

**1. flex-row :** Use flex-row to position flex items horizontally in the same direction as text

**2. flex-row-reverse :** Use flex-row-reverse to position flex items horizontally in the opposite direction

**3. flex-col :** Use flex-col to position flex items vertically

**4. flex-col-reverse :** Use flex-col-reverse to position flex items vertically in the opposite direction

**->Flex Wrap**

- Utilities for controlling the direction of flex items.

**<-> Classes:**

**1. flex-wrap :** Use flex-wrap to allow flex items to wrap

**2. flex-wrap-reverse :** Use flex-wrap-reverse to wrap flex items in the reverse direction

**3. flex-nowrap :** Use flex-nowrap to prevent flex items from wrapping, causing inflexible items to overflow the container if necessary

**-> Flex :**

- Utilities for controlling how flex items both grow and shrink.

<-> Classes:

**1. flex-1 :** Use flex-1 to allow a flex item to grow and shrink as needed, ignoring its initial size

**2. flex-auto :** Use flex-auto to allow a flex item to grow and shrink, taking into account its initial size

**3. flex-initial :** Use flex-initial to allow a flex item to shrink but not grow, taking into account its initial size

**4. flex-none :** Use flex-none to prevent a flex item from growing or shrinking:

**-> Flex Grow**

- Utilities for controlling how flex items grow.

**<-> Classes:**

**1. flex-grow-0 :** Use flex-grow-0 to prevent a flex item from growing

**2. flex-grow :**Use flex-grow to allow a flex item to grow to fill any available space

**-> Flex Shrink**

- Utilities for controlling how flex items shrink.

**<-> Classes:**

**1. flex-shrink-0 :** Use flex-shrink-0 to prevent a flex item from shrinking

**2. flex-shrink :** Use flex-shrink to allow a flex item to shrink if needed

**2. Grid**

**-> Grid Template Rows**

- Utilities for specifying the rows in a grid layout.

- Use the grid-rows-{n} utilities to create grids with n equally sized rows.

- **Responsive :** To control the rows of a grid at a specific breakpoint, add a {screen}: prefix to any existing grid-template-rows utility. For example, use md:grid-rows-6 to apply the grid-rows-6 utility at only medium screen sizes and above.

Ex : <div class="grid grid-rows-2 md:grid-rows-6 ...">

<!-- ... -->

</div>

**-> Spanning rows / columns**

- Use the row-span-{n} utilities to make an element span n rows.

- Use the col-span-{n} utilities to make an element span n cols.

**-> Grid Row Start/End**

- Use the row-start-{n} and row-end-{n} utilities to make an element start or end at the nth grid line. These can also be combined with the row-span-{n} utilities to span a specific number of rows.

Note that CSS grid lines start at 1, not 0, so a full-height element in a 3-row grid would start at line 1 and end at line 4.

**-> Grid Auto Flow**

**<-> Classes:**

**1. grid-flow-row**

**2. grid-flow-col**

**3. Justification**

**-> Justify Content**

- Utilities for controlling how flex and grid items are positioned along a container's main axis

**<-> Classes**

**1. justify-start**

**2. justify-end**

**3. justify-center**

**4. justify-between**

**5. justify-around**

**6. justify-evenly**

**-> Justify Items**

- Utilities for controlling how grid items are aligned along their inline axis.

**<-> Classes**

**1. justify-items-start**

**2. justify-items-end**

**3. justify-items-center**

**4. justify-items-stretch**

Ex: <div class="grid justify-items-start ...">

<div>1</div>

<div>2</div>

<div>3</div>

<div>4</div>

<div>5</div>

<div>6</div>

</div>

**4. Align**

**-> Align Content**

- Utilities for controlling how rows are positioned in multi-row flex and grid containers.

**<-> Classes**

**1. content-center :** Use content-center to pack rows in a container in the center of the cross axis

**2. content-start :** Use content-start to pack rows in a container against the start of the cross axis

**3. content-end :** Use content-end to pack rows in a container against the end of the cross axis

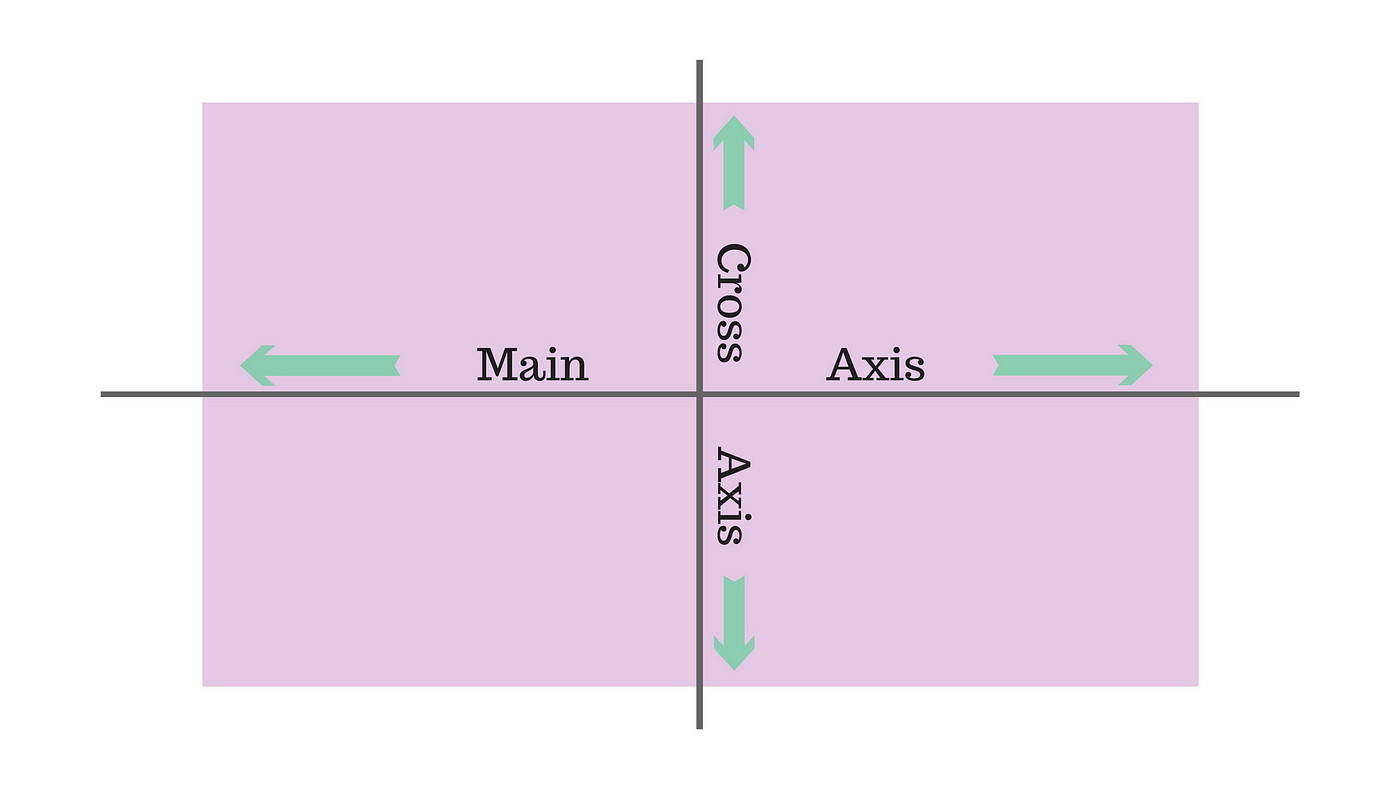
**4. content-between :** Use content-between to distribute rows in a container such that there is an equal amount of space between each line

**5. content-around :** Use content-around to distribute rows in a container such that there is an equal amount of space around each line

**6. content-evenly :** Use content-evenly to distribute rows in a container such that there is an equal amount of space around each item, but also accounting for the doubling of space you would normally see between each item when using content-around

**-> Align Items**

- Utilities for controlling how flex and grid items are positioned along a container's cross axis.



**<-> Classes**

**1. items-start**

**2. items-end**

**3. items-center**

**4. items-baseline**

**5. items-stretch**

**\*\* NOTE :**

* If we want to specify a specific pixel size we can use the syntax like h-[pixel size] or [height:50px] for height,w-[pixel size],

Here h means height,w means width etc…

Ex : h-[50px] or [height:50px] // height of 50 pixel size

* For any property containing spaces we can include \_ in the space of spaces like the below example

Ex: [box-shadow: 0\_0\_10px\_black] in tailwind css

Normally we write the above property as

box-shadow:0 0 10px black

**5. Shadows**

- Utilities for controlling the box shadow of an element.

**<-> Classes**

**-> Outer shadow :** Use the shadow-sm, shadow, shadow-md, shadow-lg, shadow-xl, or shadow-2xl utilities to apply different sized outer box shadows to an element.

**-> Inner Shadow :** Use the shadow-inner utility to apply a subtle inset box shadow to an element. This can be useful for things like form controls or wells.

**-> We can also put the shadow color like class="shadow shadow-red-100" like that**

**6. Borders**

**-> Border radius**

- Utilities for controlling the border radius of an element.

**- Rounded Corners :** Use utilities like **.rounded-sm, .rounded, or .rounded-lg** to apply different border radius sizes to an element.

**- Pills and Circles :** Use the **rounded-full** utility to create pills and circles.

**- No rounding : Use rounded-non**e to remove an existing border radius from an element.

- **Rounding sides separately :** Use rounded-{**t|r|b|l}{-size?}** to only round one side of an element.

- **Rounding corners separately : Use rounded-{tl|tr|br|bl}{-size?}** to only round one corner an element.

**-> Border Width**

- Utilities for controlling the width of an element's borders.

- **All sides :** Use the **border, .border-0, .border-2, .border-4, or .border-8 utilities** to set the border width for all sides of an element.

- **Individual sides :** Use the **border-{side}, .border-{side}-0, .border-{side}-2, .border-{side}-4, or .border-{side}-8 utilities** to set the border width for one side of an element.

**here side means t(top),b,l,r**

**-> Border Color**

- Control the border color of an element using the **border-{color} utilities.**

- Control the opacity of an element’s border color using the **border-opacity-{amount} utilities.**