

# Basic Sizing

• Size can be measured Relatively or Absolutely

• There are many options, the useful ones are

- **px** – size in pixels

יחידה בסיבית שנמצאת במסך מסך מורכב ממלא נקודות שנקראות פיקסלים

- **rem** - the default font-size as user have set it (usually  
html 1rem=16px גודל ביחס ל16px~)

- **em** – the current default font-size

1em=16px (if modified by parent elements)

אם אין לאבא שלו גודל הולך לhtml

- **%** - usually this means a percent of the parent dimensions

$14px/16=0.875rem/em$

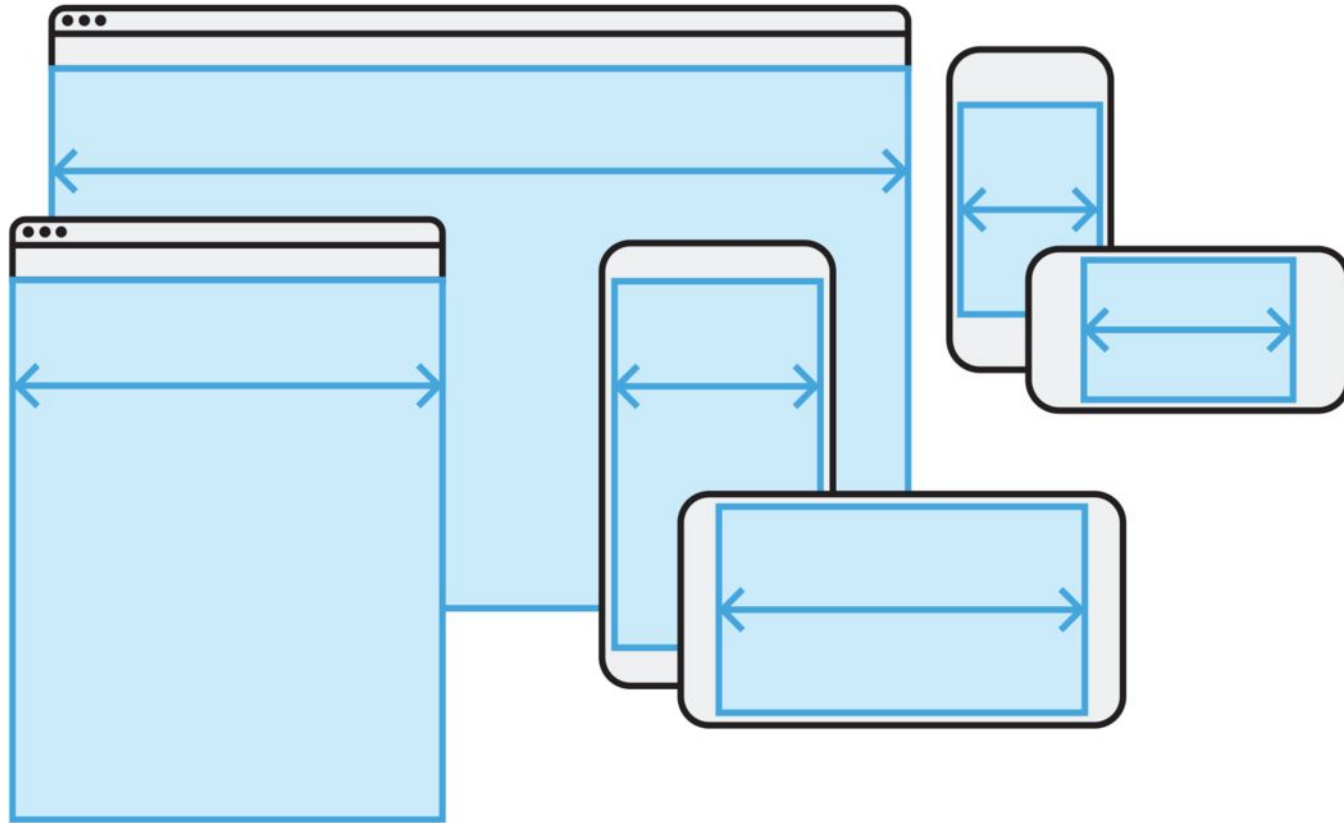
$32px/16=2rem/em$

אם כבד ראיה יגדיל את המסך אז זה יחלק לפני היחס של הגודל למשל אם גודל מסך 23 פיקסל  
ואלמנט מוגדר ב0.5 rem

אז זה יהיה בפועל בגודל של 11.5 פיקסלים

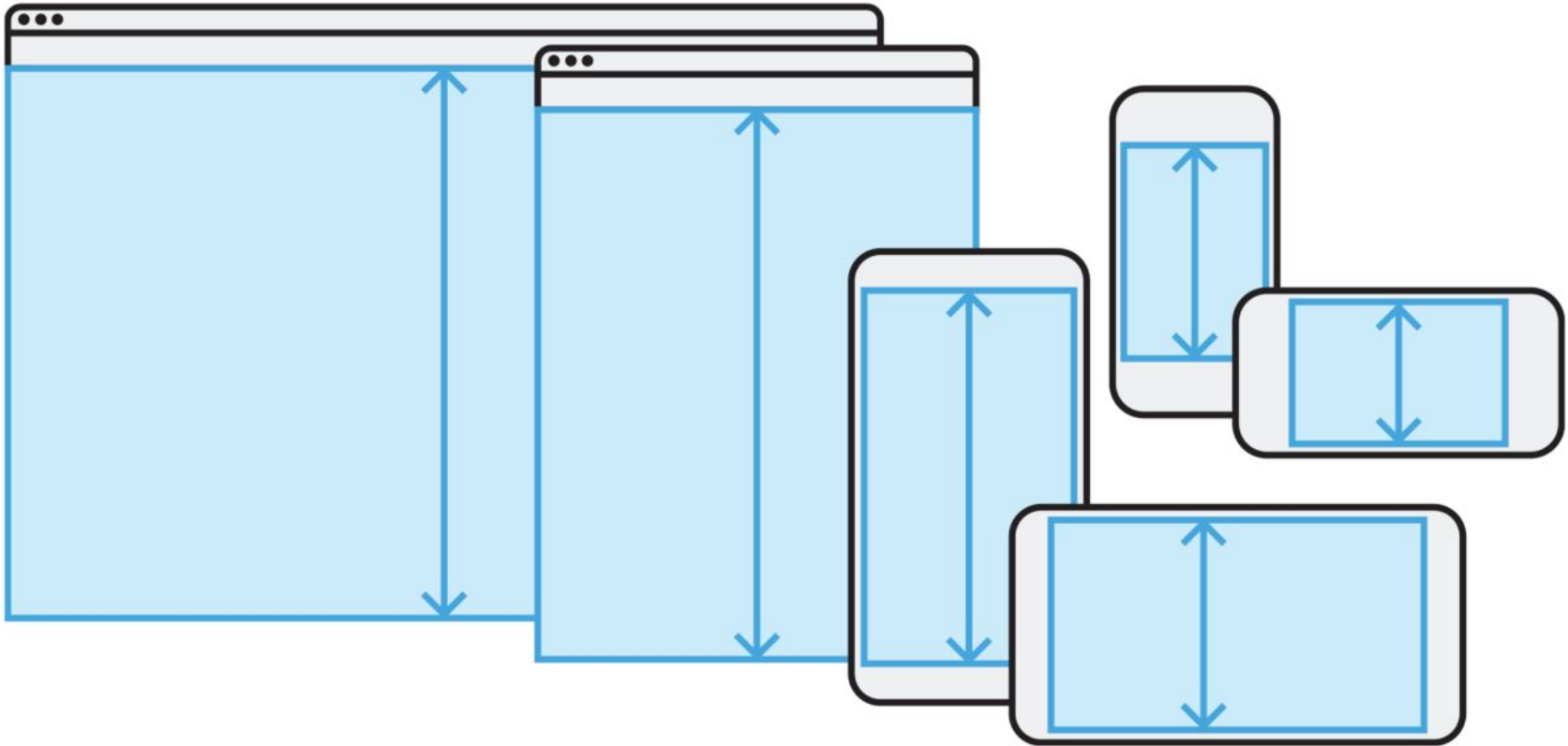
# Viewport Units

`100vw = 100% of viewport width`



# Viewport Units

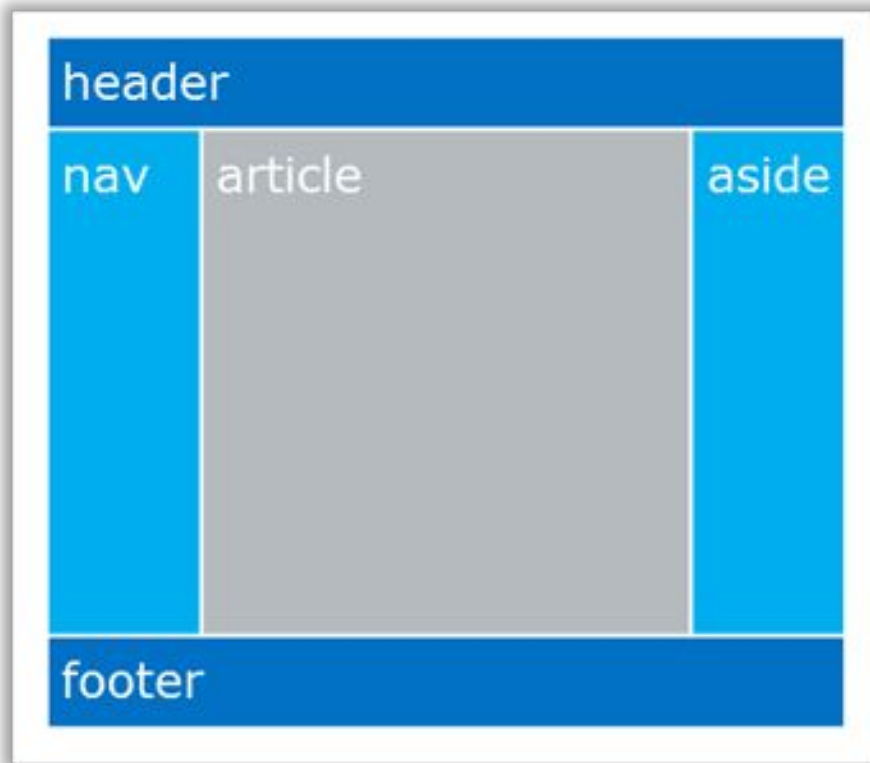
`100vh = 100% of viewport height`



**Link:favicon**

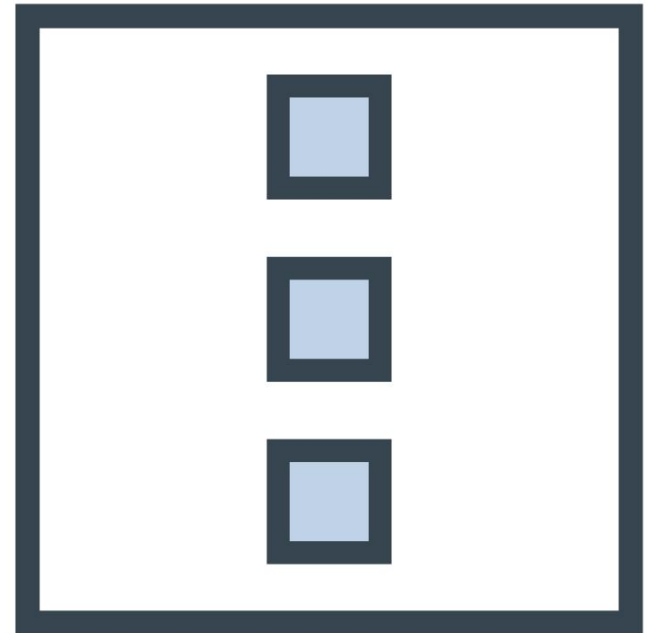
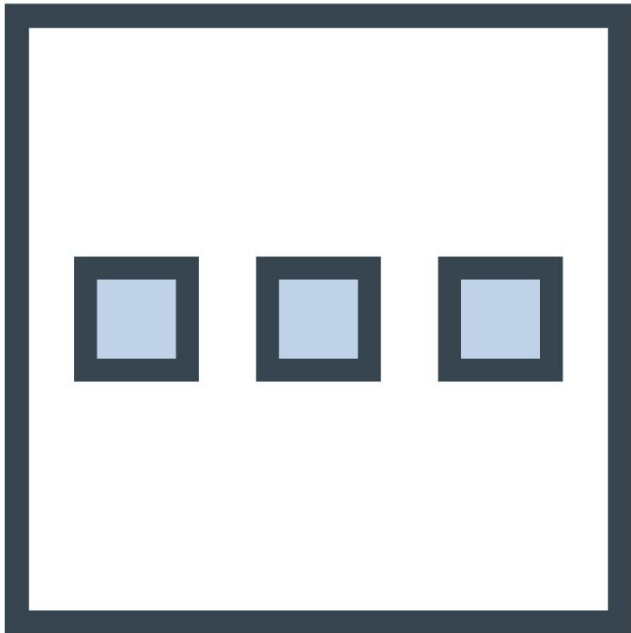
# Layouting with flex

**Flexbox** is the most common way to layout websites today. It is a very powerful tool .to master



# ~~Flexbox~~

In the next slides, we will call:  
the **Flex Container** - Parent  
and the **Flex Item** Child

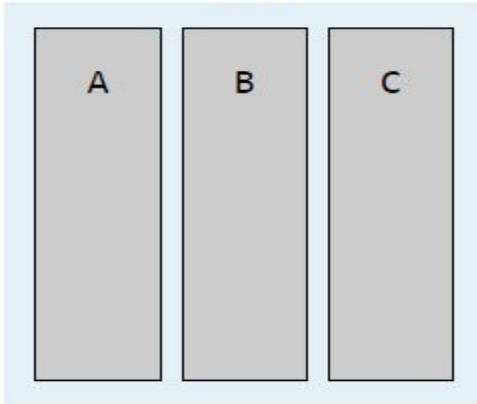


## Parent

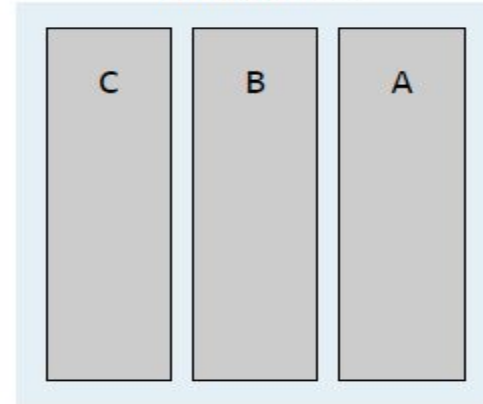
# Flex-direction

תכונה שבעזרתם אני יכולה לשלוט באיזה כיוון  
האלמנטים יסתדרו **מכתיבה לי מה הציר המרכזי**  
שעליו נעבוד

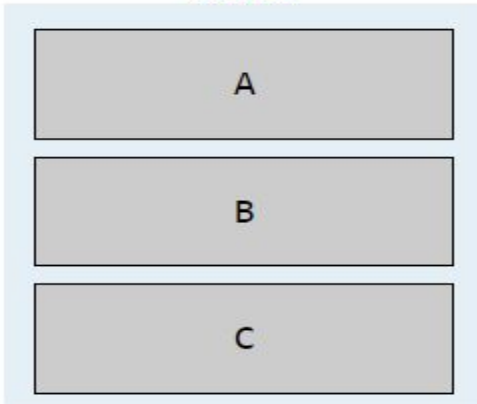
row



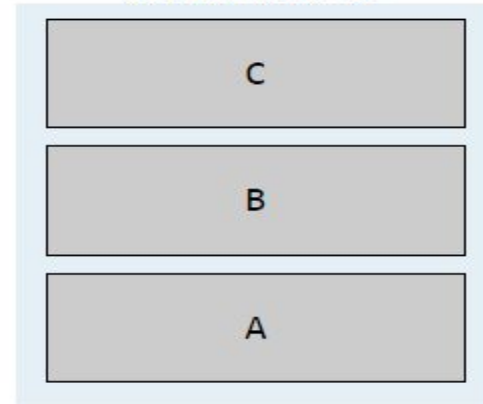
row-reverse



column



column-reverse

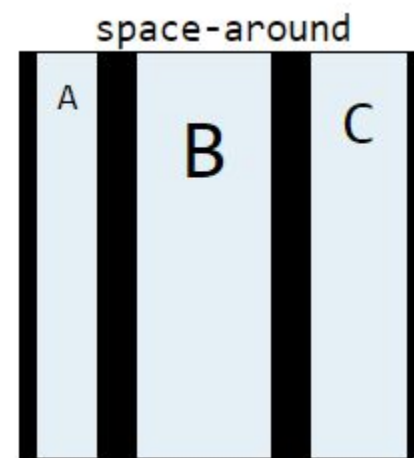
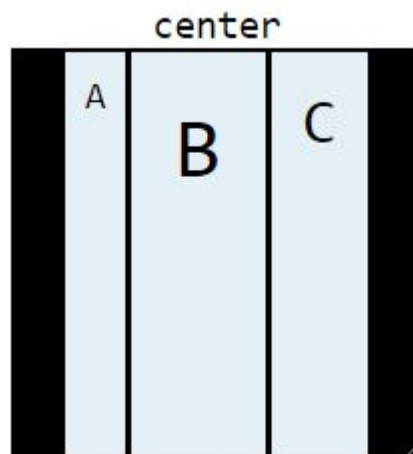
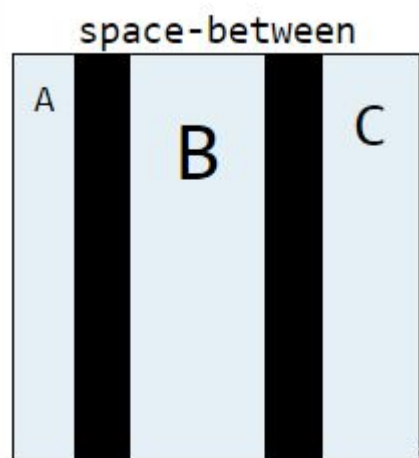
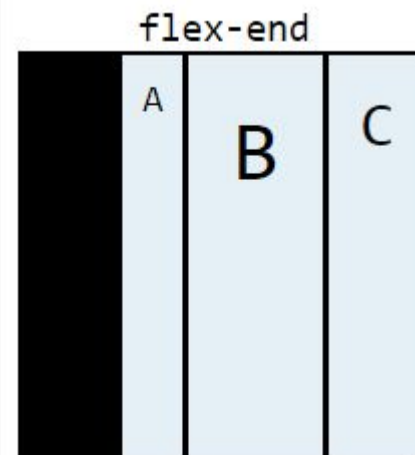
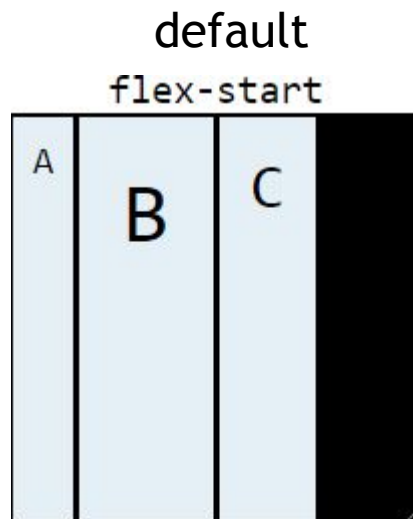
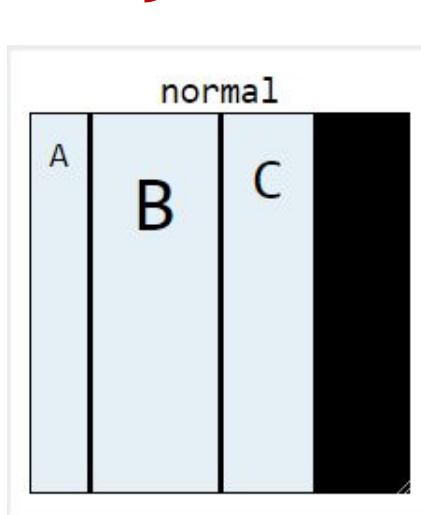


Default=flex-direction: row

# Justify-content

תכונה שבעזרתם אני יכולה למקם אלמנטים על הציר המרכזי

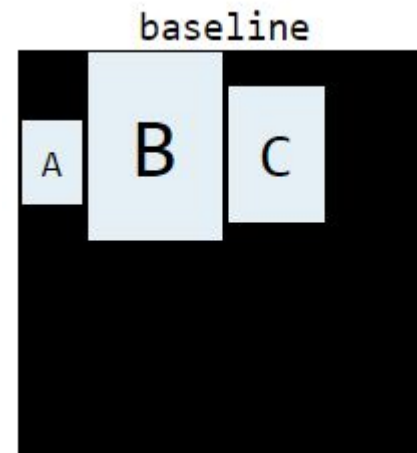
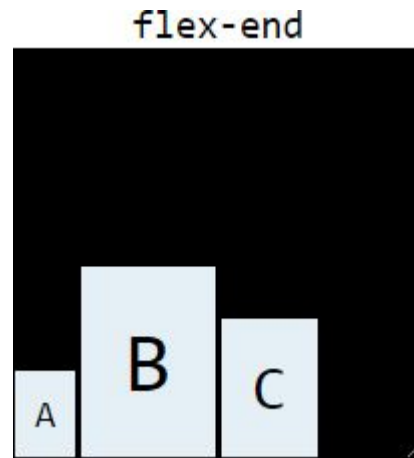
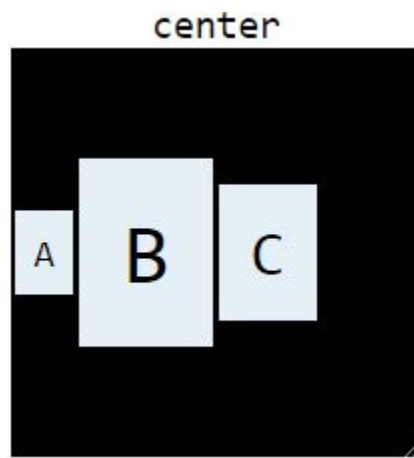
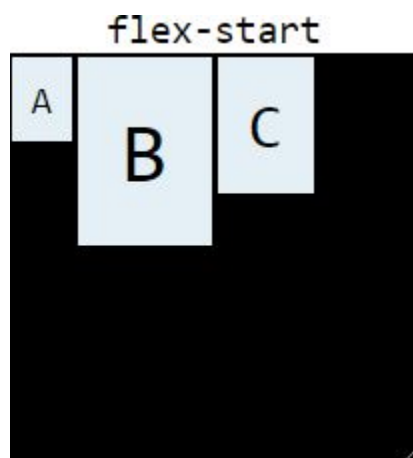
שאני נמצאת כרגע תמונות לפי ציר X





# align-items

תכונה שמשפיעה על אמנטיים  
בציר משני, מזיז את כל הבלוק



Default=align-items: stretch

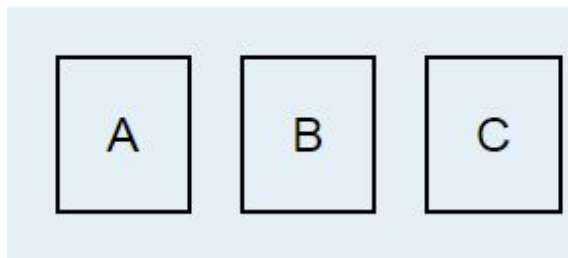
נמתח על הציר הנגדי כמה שיכול  
אפשר לתת לכל אלמנט גובה/רוחב כדי לבטל את ההתנהגות  
תמונות לפי ציר X

# Flex-wrap

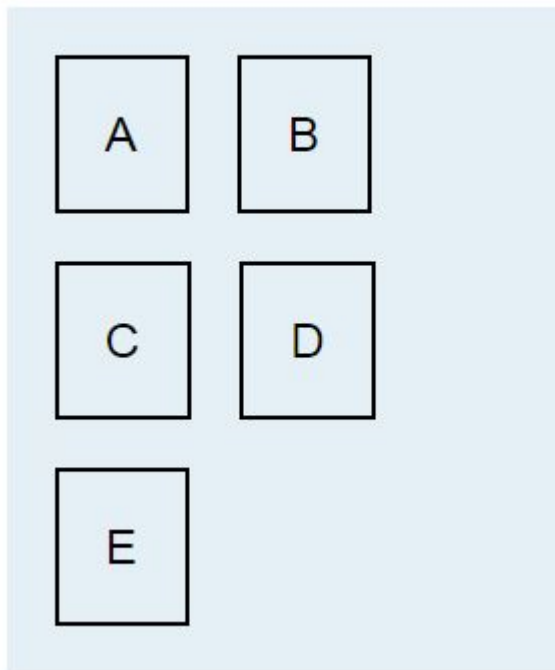
עוזר לי להוריד אלמנטים אחד מתחת לשני  
בהקטנת מסך עוזר לרספונסיביות

Default

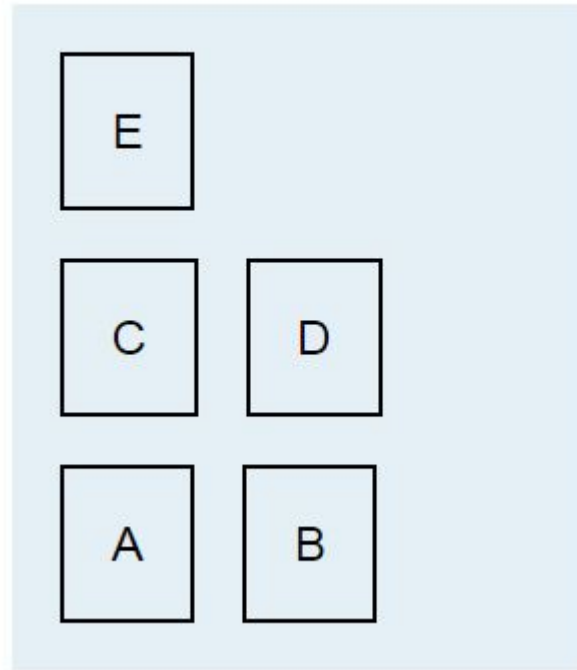
`flex-wrap: nowrap`



`flex-wrap: wrap`

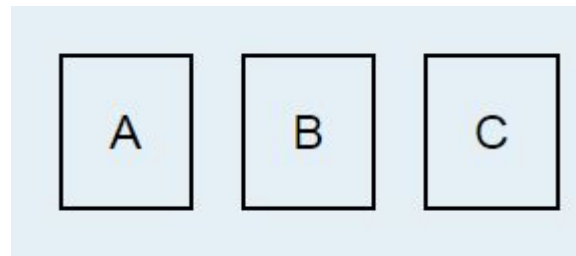


`flex-wrap: wrap-reverse`



# flex-flow (shorthand)

property is a shorthand property for: `flex-direction`  
and `flex-wrap`



`flex-flow: row nowrap;`

`flex-flow: column wrap`

`flex-flow: column-reverse wrap-reverse`

# flex-basis

תכונה שמדברת על הגודל ההתחלתי של flex-items מתייחס לציר המרכזי

This property determines the size of the **content-box** unless specified otherwise using **box-sizing**.

**flex-basis: 5em;**

**flex-basis: 60px;**

**;Flex-basis:150px**

אם ציר מרכזי יהיה X זה ישפיע על הרוחב  
אם ציר מרכזי יהיה Y זה ישפיע על הגובה

# gap

- ריווח בין flex-items
- `gap:20px`; ריווח בין טורים ושורות
- `row-gap: 20px`; ריווח 20 פיקסל בין שורות
- `column-gap: 40px`; ריווח 40 פיקסל בין עמודות
- (shorthand)
- `gap:20px 40px`;
- 20 פיקסל ריווח בין שורות ו40 פיקסל בין הטורים

# flex-grow

תכונה שמשפיעה על הילדים על flex-item היא  
עוזרת להחליט כמה רווח שנשאר מהקונטינר הוא  
יקח

איך לחלק שטח הפרש  
תמיד ביחס לאייטם השני שגם יש לו flex-grow

Default:0

flex-grow:

0.5

1

2

flex-grow: 2

flex-grow: 0.5

# flex-shrink

עובד על כיווציות

עובד ביחס לאחרים שיש להם גם תכונה של flex-shrink

Default:0

flex-shrink:

0.5

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at purus vitae ipsum hendrerit vulputate quis vitae risus.

1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at purus vitae ipsum hendrerit vulputate quis vitae risus.

3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at purus vitae ipsum hendrerit vulputate quis vitae risus.

;**flex-shrink: 3**

;**flex-shrink: 0.5**

# flex (shorthand)

The **flex** property is a shorthand property that sets `.flex-grow`, `flex-shrink`, and `flex-basis`

אם אני כותבת רק מספר בלי שום ערך מתייחס `flex-grow`

`Flex:2`

אם אני כותבת רק מספר עם יחידת מידה מתייחס `flex-basis`

`;Flex:10em`

`Flex:30px`

שני ערכים 1 בלי יחידת מידה והשני עם יתיחס ל:

`flex-grow | flex-basis`

`Flex: 1 50px`

שני ערכים בלי יחידת מידה מתייחס ל `flex-grow | flex-shrink`

`Flex:2 2`

`/* flex-grow | flex-shrink | flex-basis :`



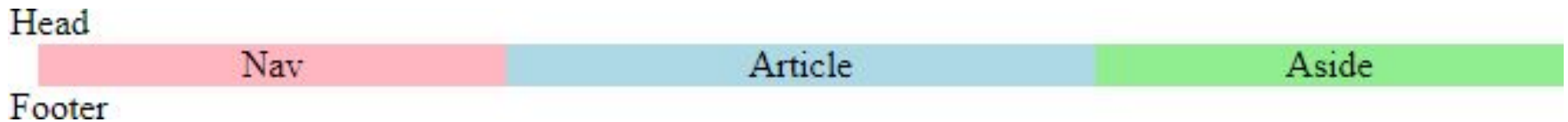
# Order

תכונה שמאפשרת לי להחליט מה סדר האלמנטים ,

לשנות את הסדר הhtml

הסדר הוא ביחס לאחרים

מי שאין לו את התכונה זה יהיה לפי הסדר שלו בHTML



Default:order:0

**Order:1** <= הראשון מבין מי שיש לו את התכונה

**order: -1** <=; תיקח אותו ראשון

Child

# align-self try [here](#)

תכונה שמאפשרת לי להחליט איפה כל אחד

מהילדים ימקום ביחס לציר הנגדי

.The property doesn't apply to block-level boxes, or to table cells

**align-self:** center; /\* Put the item around the center \*/

**align-self:** flex-start; /\* Put the flex item at the start \*/

/\* **align-self:** flex-end; /\* Put the flex item at the end

...Lets make it

