

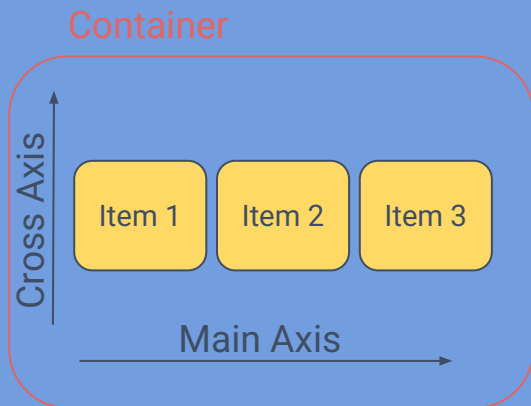


# CSS Flexbox

+ מהבסיס ליישום מעשי

# מה זה בעצם Flexbox?

- מודל פריסה (Layout Model) חד-ממדי.
- מיועד לסידור אלמנטים בשורה או בטור.
- אינטואיטיבי, גמיש וחזק.



## מושגי יסוד:

- Flex Container (ההורה)
- Flex Items (הילדים)
- Main Axis (הציר הראשי)
- Cross Axis (הציר הצולב)

# ה-Flex Container

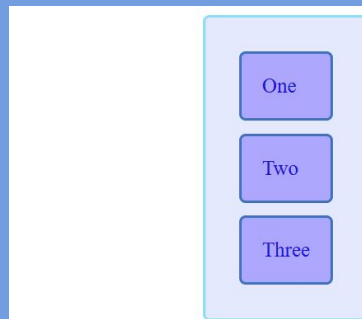
נתחיל בתכונות שחלות על אלמנט ההורה.

• `display: flex`

- המתג ש"מדליק" את כל הקסם.
- ברגע זה, הילדים הישירים הופכים ל-flex Items.

• `flex-direction`

- קובע את כיוון הציר.
- `row`: (ברירת מחדל): שורה, משמאל לימין.
- `column`: טור, מלמעלה למטה.
- `row-reverse` | `column-reverse`



```

}
.container {
  border: 2px solid #8ae0f3;
  border-radius: 5px;
  background-color: #e6e8ff;
  padding: 1em;
}
div {
  display: block;
  user agent

```

# יישור על הציר הראשי



- `justify-content`
  - `flex-start` (התחלה) | `flex-end` (סוף)
  - `center` (מרכז)
  - `space-between` (רווח בין האלמנטים)
  - `space-around` (רווח סביב האלמנטים)
  - `Space-evenly` (רווח אחיד לחלוטין)



```
✓ border: 2px solid #8ae0f3;  
✓ border-radius: 5px;  
✓ background-color: #e6e8ff;  
✓ padding: 1em;  
✓ width: 500px;  
✓ display: flex;  
}
```



## תרגול 1: navbar

**המשימה :** סדרו את הלוגו בצד שמאל ואת רשימת הקישורים בצד ימין.

**קישור :** לחצו כאן.

**תוצר נדרש:**

MyLogo

Home About Products Contact

# יישור על הציר הנגדי

`align-items`

- קובע את יישור האלמנטים לאורך ה-Cross Axis.
- `stretch` (ברירת מחדל): מותח את האלמנטים לגובה/רוחב מלא.
- `flex-start` (התחלה) | `flex-end` (סוף)
- `center` (מרכז)
- `baseline` (מיישר לפי קו הבסיס של הטקסט)



The diagram shows a light blue rectangular container. Inside the container, on the left side, are three vertical purple rectangles. The first rectangle is labeled 'One', the second 'Two', and the third 'Three'. They are all aligned to the top of the container.

```
.container {  
  border: 2px solid #8ae0f3;  
  border-radius: 5px;  
  background-color: #e6e8ff;  
  padding: 1em;  
  width: 500px;  
  height: 200px;  
  display: flex;  
}
```

## תרגול 2: מרכז מושלם

**המשימה :** למרכז ריבוע בדיוק באמצע העמוד (אופקית ואנכית).

**קישור :** [לחצו כאן](#).



# Flex Items - שליטה על כל אלמנט בנפרד

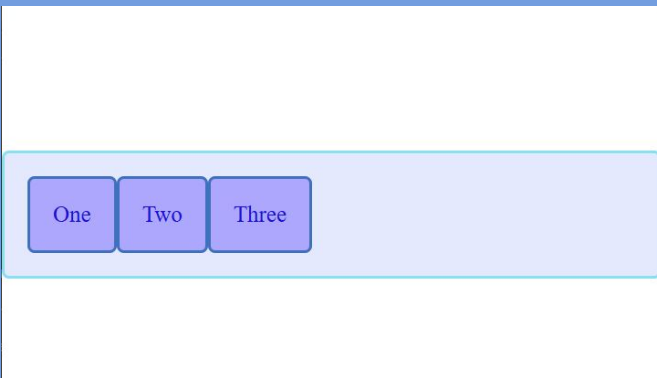
עכשיו נדבר על תכונות שחלות על אלמנטי הילדים.



# גמישות, גדילה וקטינה

`flex-grow` | `flex-shrink` | `flex-basis` •

- קובע את יישור האלמנטים לאורך ה-Cross Axis.
- `flex-basis` הגודל ההתחלתי של האלמנט.
- `flex-grow` היכולת של אלמנט לגדול ולתפוס מקום פנוי (ברירת מחדל: 0).
- `flex-shrink` היכולת של אלמנט להתכווץ כשאין מקום (ברירת מחדל: 1).
- קיצור אפשרי (לרוב נשתמש בו) `flex: <grow> <shrink> <basis>`



```
.box {  
  border: 2px solid #4072bd;  
  border-radius: 5px;  
  background-color: #aea8ff;  
  padding: 1em;  
  color: #1c0fd9;  
}  
  
div {  
  user agent s
```

## תרגול 3: תבנית עמוד

**המשימה :** צרו תבנית עם navbar בגובה התחלתי של 100px, סרגל צד ברוחב קבוע של 250px, ותוכן מרכזי שתופס את כל שאר המקום.

**קישור :** [לחצו כאן](#).

תוצר נדרש:

Navbar

Sidebar

Main Content



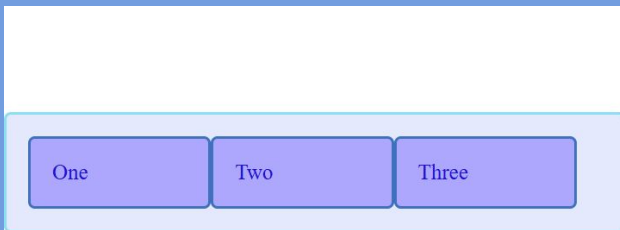
# גלישת שורות ורווחים

`flex-wrap: wrap`

- מאפשר לאלמנטים לרדת לשורה חדשה בשאין מספיק מקום.
- הופך את הפריסה לרספונסיבית כמעט ללא לגמרי.

`gap: <value>`

- הדרך המודרנית והנוחה ליצור רווחים אחידים בין האלמנטים.
- הרבה יותר טוב מ `margin` במקרה הזה.



```
.container {  
  border: ▶ 2px solid #8ae0f3;  
  border-radius: ▶ 5px;  
  background-color: #e6e8ff;  
  padding: ▶ 1em;  
  width: 500px;  
  display: flex;  
}  
  
div {
```

*user ag*

# תרגול 4: גלריית כרטיסיות

**המשימה :** לסדר את הכרטיסיות כך שיגלשו בין שורות ויתאימו את עצמם לרוחב המסך (הגודל המינימלי של כל כרטיסיה הוא 150px).

**קישור :** לחצו כאן.

תוצר נדרש:

## Our Awesome Products

Card 1

Card 2

Card 3

Card 4

Card 5

Card 6



# סיכום

● אז מה היה לנו?

- למדנו על ההבדל בין `Container` ו-`Items`.
- יישרנו אלמנטים על שני הצירים (`justify-content`, `align-items`).
- יצרנו פריסות גמישות עם `flex-grow` ו-`flex-basis`.
- בנינו רכיבים רספונסיביים עם `flex-wrap` ו-`gap`.
- בשורה התחתונה: Flexbox הוא הכלי המרכזי לפריסה חד-ממדית.



תודה על ההקשבה



# לינקים שימושיים

## ● פתרון התרגילים:

○ תרגיל 1 - [באן](#)

○ תרגיל 2 - [באן](#)

○ תרגיל 3 - [באן](#)

○ תרגיל 4 - [באן](#)

## ● מקורות להמשך למידה:

○ משחק מהנה ומלמד FlexBox Froggy - [באן](#)

○ MDN Web Docs אתר מלא בדוגמאות, לינק

ישירות לחלק שמדבר על FlexBox - [באן](#)

