

Flexbox

קורס : צד לקוח

מרצה: יעל סלע זעירא

Flexbox הקדמה

- Flexbox נותן לנו שימוש נוח ליצירת layout (לבטח כשמדובר ב layout חד מימדי של שורות או עמודות) .
- לעיתים ברצוננו להמנע כמה שניתן מתלות בספריות ונכסים חיצוניים (כדוגמאת bootstrap) ואנו מעוניינים להשתמש רק ביכולות של CSS – flexbox הוא המענה עבורנו.

מודל ה flexbox

שני סוגים של "קופסאות" :

Flex container .1



“FLEX CONTAINER”

Flex items .2



“FLEX ITEMS”

display: flex;

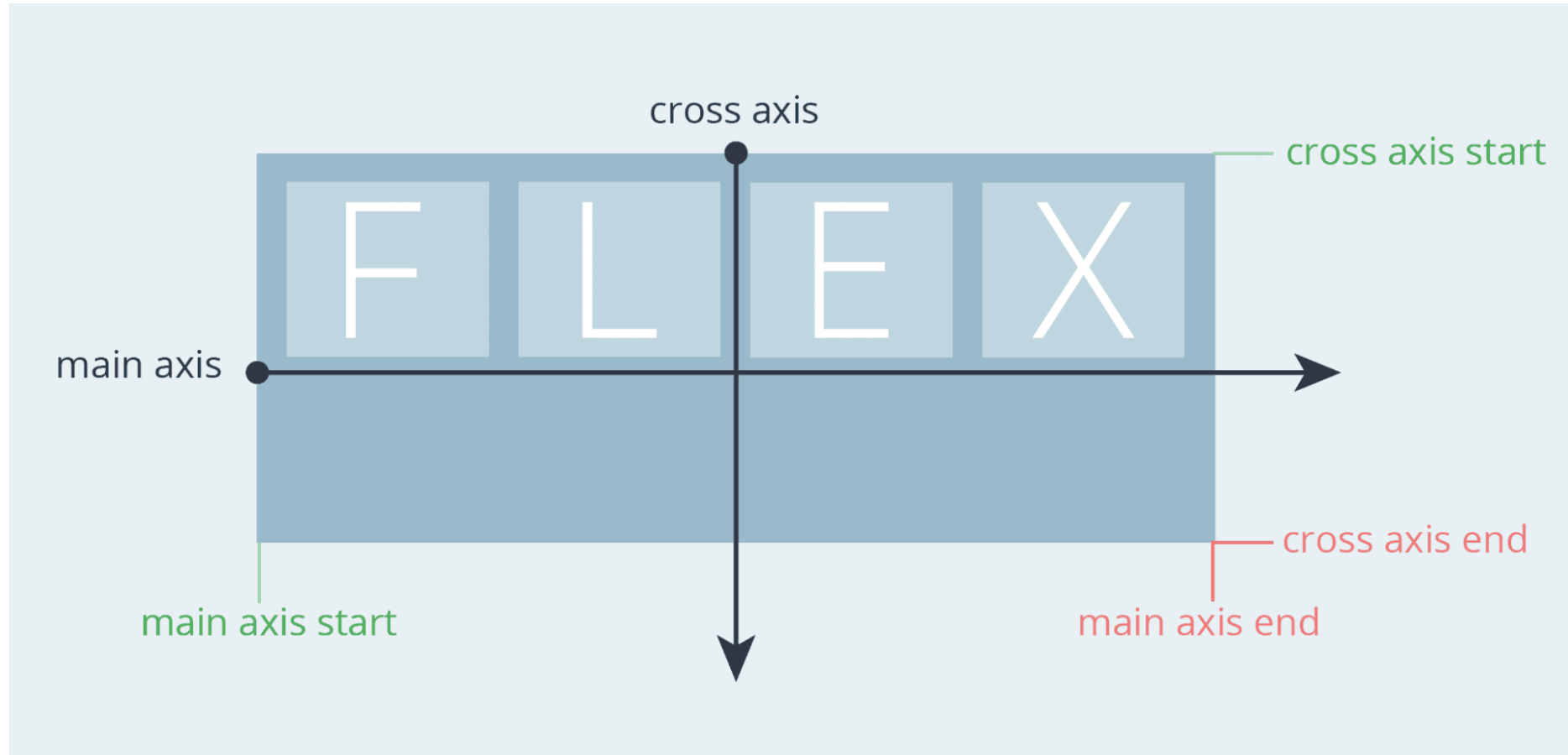
```
<head>
<style>
.flex-container {
    display: flex;
}
</style>
</head>
<body>
    <div class="flex-container">
        <div>Flex Item</div>
        <div>Flex Item</div>
        <div>Flex Item</div>
    </div>
</body>
```

- על ידי הוספת התכונה `display: flex;` לאלמנט מסויים.

- הוא יהפוך ל Flex container וילדיו ל Flex items

Flex ItemFlex ItemFlex Item

Main-Axis and Cross-Axis



מיקום ויישור אלמנטים בציר ה-Y האנכי

`align-items`

מאפשרת לנו "לשחק" עם האלמנטים אנכית בציר ה-y

`align-items: center;`

```
<head>
<style>
.flex-container {
display: flex;
border: 1px solid #1a232d;
height: 200px;
align-items:center
}
</style>
</head>
<body>
<div class="flex-container">
<div>Flex Item</div>
<div>Flex Item</div>
<div>Flex Item</div>
</div>
</body>
```

Flex ItemFlex ItemFlex Item

align-items:flex-end

Flex ItemFlex ItemFlex Item

מיקום ויישור אלמנטים בציר ה- X האנכי

`justify-content`

מאפשרת לנו "לשחק" עם האלמנטים אופקית בציר ה- X

`justify-content: flex-start;`

* ב *Flex Container* אלמנטים מסתדרים בשורה ובתחילת ציר ה- X מכיוון וערך ברירת המחדל של התכונה הוא `flex-start`

```
<head>
<style>
.flex-container {
  display: flex;
  border: 1px solid #1a232d;
  height: 200px;
  justify-content: space-between;
}
</style>
</head>
<body>
<div class="flex-container">
<div>Flex Item 1</div>
<div>Flex Item 2</div>
<div>Flex Item 3</div>
</div>
</body>
```

Flex Item 1

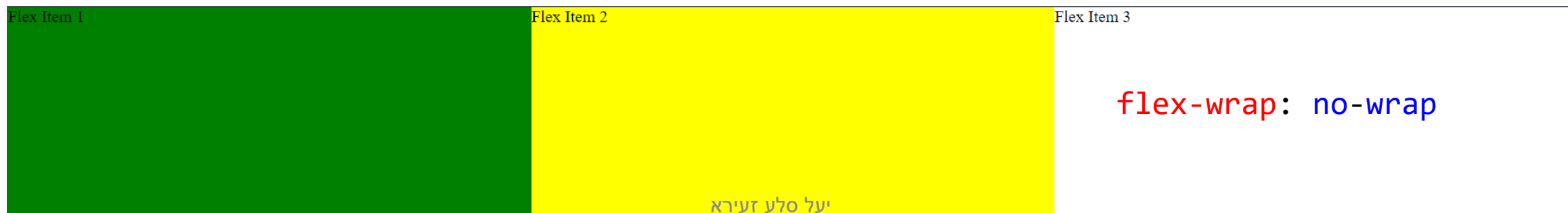
Flex Item 2

Flex Item 3

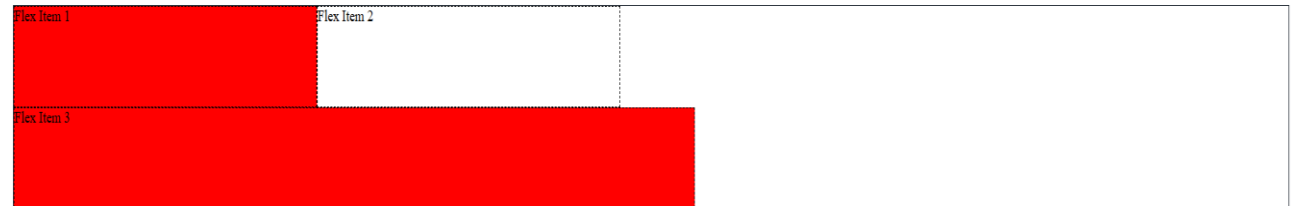
flex-wrap

• **flex-wrap** – כברירת מחדל אלמנטים לא יגלשו לשורה חדשה אם אין להם מקום בקונטיינר המכיל אותם.

• אם נרצה שהאלמנטים יגלשו לשורה חדשה בקונטיינר, יש לשים **flex-wrap: wrap;**

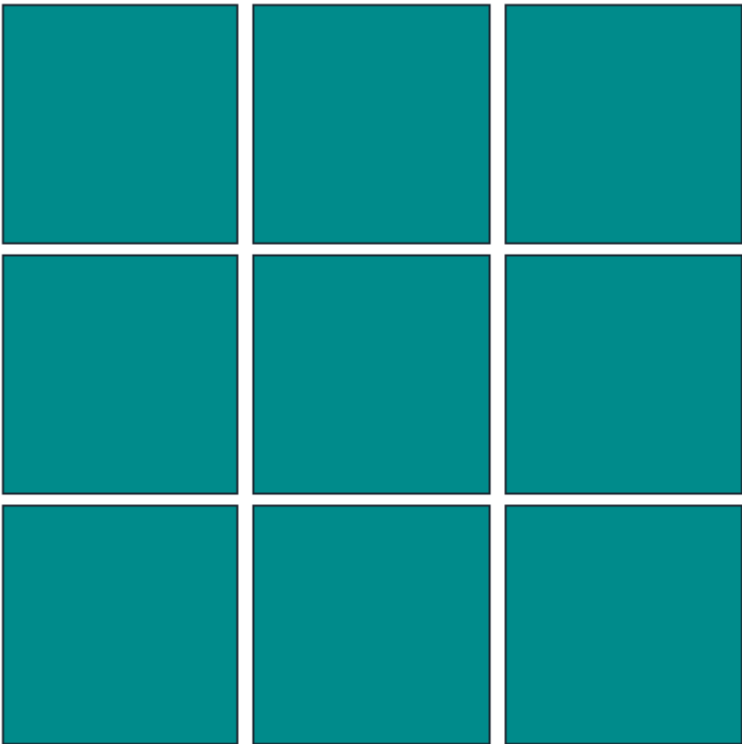


```
<head>
  <style>
    .flex-container {
      display: flex;
      border: 1px solid #1a232d;
      height: 200px;
      flex-wrap: wrap;
    }
    .flex-container > div {
      width: 400px;
      border: 1px dashed black;
    }
    .flex-container > div:last-child {
      width: 900px;
    }
    .flex-container > div:nth-child(odd) {
      background-color: red;
    }
  </style>
</head>
<body>
  <div class="flex-container">
    <div>Flex Item 1</div>
    <div>Flex Item 2</div>
    <div>Flex Item 3</div>
  </div>
</body>
```



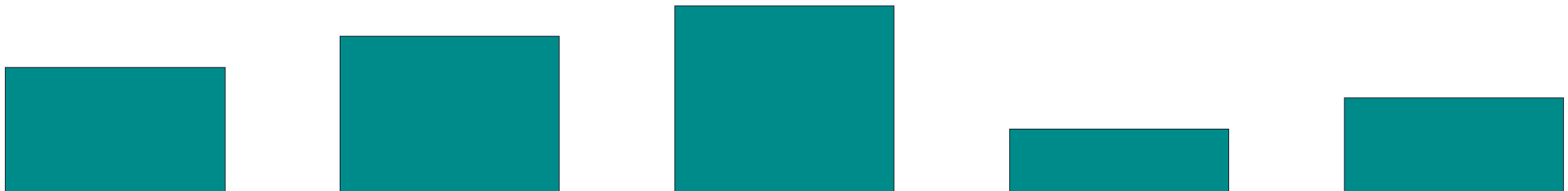
תרגיל כיתה 1

צייר את הגריד הבא על ידי שימוש ב `display: flex`



תרגיל כיתה 2

צייר את הסכמה הבאה על ידי שימוש ב `display: flex`



תכונות של אלמנט Flex Item

align-self

מאפשרת לשנות מיקום של אלמנט ספציפי בתוך הקונטיינר.

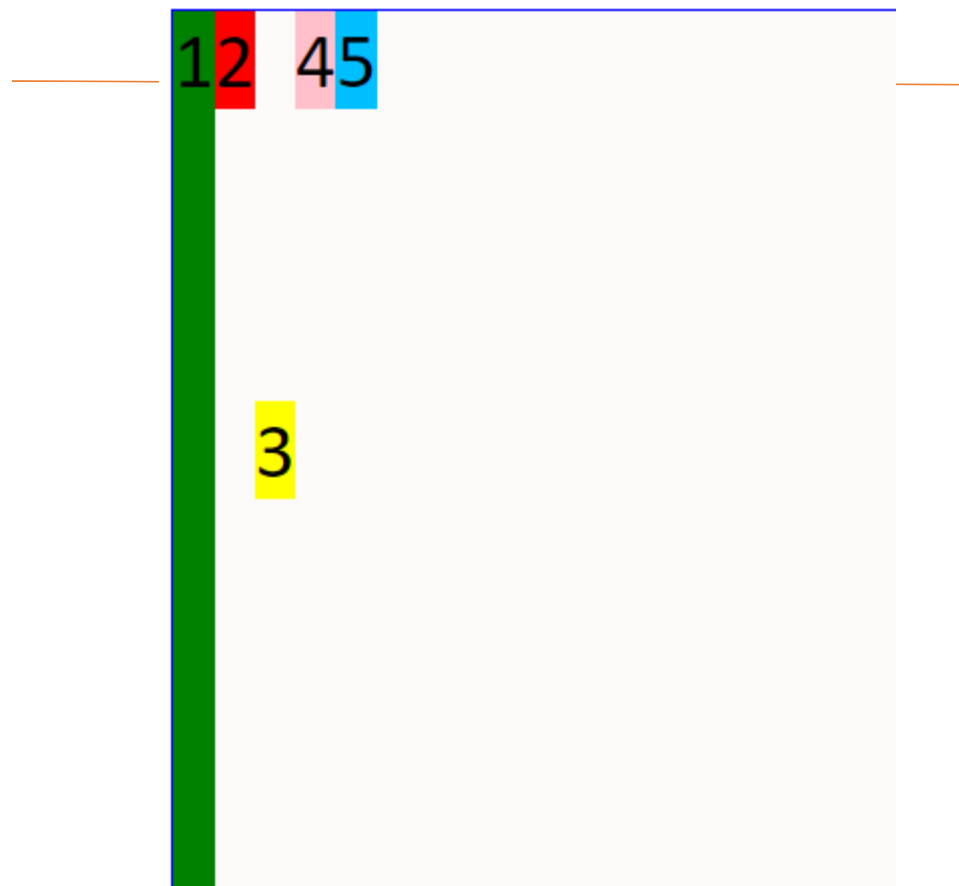
מאפשרת לנו לשבור את הגדרת ברירת המחדל הדיפולטיבית של align-items הקיימת לקונטיינר עבור אלמנט ספציפי בקונטיינר זה.

[flexItemExample.html](#)

```

<head>
  <style>
    .main-container {
      width: 100%;
      height: 350px;
      border: 1px solid blue;
      background: #fbfaf8;
    }
    .flex-wrapper {
      display: flex;
      align-items: flex-start;
    }
    .flex-wrapper > div:nth-child(1) {
      align-self: stretch;
    }
    .flex-wrapper > div:nth-child(3) {
      align-self: center;
    }
  </style>
</head>
<body>
  <div class="main-container flex-wrapper">
    <div style="background-color: green;">1</div>
    <div style="background-color: red;">2</div>
    <div style="background-color: yellow;">3</div>
    <div style="background-color: pink;">4</div>
    <div style="background-color: deepskyblue;">5</div>
  </div>
</body>

```



תכונות של אלמנט Flex Item

flex-grow

באמצעות התכונה flex-grow ניתן לשלוט על כמות השטח שאלמנט ספציפי יתפוס ביחס לאחרים.

* מכיוון והערך הוא יחסי, אם תגדירו את התכונה `flex-grow: 200` לכלל האלמנטים יהיה זה בדיוק כאילו קבעתם את הערך של כולם ל-1. או אם תתנו ערך של `flex-grow` רק לאלמנט אחד הוא יתפוס את השטח שהגדרתם לו אל מול היחס הדיפולטיבי שניתן לשאר האלמנטים.

```

<html>
<head>
<style>
#main {
width: 350px;
height: 100px;
border: 1px solid #c3c3c3;
display: flex;
}
#main div:nth-of-type(1) {
flex-grow: 1;
}
#main div:nth-of-type(2) {
flex-grow: 3;
}
#main div:nth-of-type(3) {
flex-grow: 1;
}
#main div:nth-of-type(4) {
flex-grow: 1;
}
#main div:nth-of-type(5) {
flex-grow: 1;
}
</style>
</head>
<body>
<h1>The flex-grow Property</h1>
<div id="main">
<div style="background-color:coral;"></div>
<div style="background-color:lightblue;"></div>
<div style="background-color:khaki;"></div>
<div style="background-color:pink;"></div>
<div style="background-color:lightgrey;"></div>
</div>
<p><b>Note:</b> Internet Explorer 10 and earlier versions do not support the flex-grow property.</p>
</body>
</html>

```

The flex-grow Property



Note: Internet Explorer 10 and earlier versions do not support the flex-grow property.

תרגיל כיתה

צור את התפריט הבא על ידי שימוש ב flex, בנוסף במעבר עכבר ישתנה הלינק בתפריט ל bold וישנה צבע



```
<div class="container">
  <div><a href="/">Home</a></div>
  <div><a href="/about">About</a></div>
  <div><a href="/blog">Blog</a></div>
  <div><a href="/jobs">Careers</a></div>
  <div><a href="/contact">Contact Us</a></div>
</div>
```