

CSS3 מודל התיבה הגמישה - flex box model

מודל התיבה הגמישה שמשמש בתקן CSS3 נועד לענות על שני צרכים עיקריים בעיצוב דפי אינטרנט:

- הצורך למקם אלמנטים זה לצד זה בקלות,
- הצורך להתאים את גודלם של אלמנטים על המסך לממשקים שונים (ממסכים בעלי מסך רחב ועד לסמארטפונים)

עד עכשיו ב-CSS היינו צריכים להשתמש בהצפת אלמנטים (float) כדי לסדר אלמנטים ועמודות האחד לצד השני, בתקן CSS3 נוכל להשתמש בתכונה חדישה ומתקדמת **box-orient** שהיא ידידותית בהרבה למטרה זו. בנוסף, נוכל להשתמש בתכונה **box-flex** כדי להתאים ביתר קלות את גודלם של אלמנטים בדף לגודלו של ההתקן שבו אנחנו משתמשים.

כדי להתחיל לעבוד עם ערכת התיבה (-webkit-box) שמציע לנו Chrome אנחנו חייבים לעשות את שני הדברים הבאים:

ברמת ה-HTML, נגדיר אלמנט אב כלשהו ומספר אלמנטים בנים שישיבו בתוכו. בדוגמת הקוד הבאה, אנחנו מייצרים שלוש פסקאות שיושבות בתוך אלמנט אב:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>

  </style>
</head>
<body>
  <div id="wrapper">
    <p class="column">עמודה ראשונה</p>
    <p class="column">עמודה שניה</p>
    <p class="column">עמודה שלישית</p>
  </div>
</body>
</html>
```

ברמת ה-CSS, נגדיר בתוך האלמנט האב :

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
    #wrapper {
      width: 100%;
      display: -webkit-box;
    }
  </style>
</head>
<body>
  <div id="wrapper">
    <p class="column">עמודה ראשונה</p>
    <p class="column">עמודה שניה</p>
    <p class="column">עמודה שלישית</p>
  </div>
</body>
</html>
```

```

    </style>
</head>
<body>
    <div id="wrapper">
        <p class="column">עמודה ראשונה</p>
        <p class="column">עמודה שניה</p>
        <p class="column">עמודה שלישית</p>
    </div>
</body>
</html>

```

סידור אלמנטים זה לצד זה

כעת נוסיף מעט הגדרות בחלק העיצוב:

```

<!DOCTYPE html>
<html>
<head>
    <title></title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <style>
        #wrapper {
            border: 1px solid black;
            width: 100%;
            display: -webkit-box;
            -webkit-box-orient: horizontal;
        }

        .column {
            border: 1px solid black;
            background-color: yellow;
            padding: 10px;
            margin: 7px;
        }
    </style>
</head>
<body>
    <div id="wrapper">
        <p class="column">עמודה ראשונה</p>
        <p class="column">עמודה שניה</p>
        <p class="column">עמודה שלישית</p>
    </div>
</body>
</html>

```

- נשים לב, שהוספנו webkit לאלמנט האב wrapper, ולא לילדים. (column). התוצאה שהעמודות מסודרות אחת לצד השנייה.
- הערכים ש `-webkit-box-orient` יכול לקבל הם: `horizontal` לסידור האלמנטים זה לצד זה או `vertical` לסידור האלמנטים האחד מעל לשני.

מרכז האלמנטים בדף

אלמנטים של CSS ניתן למרכז לרוחב הדף (באופן הוריזנטלי) ולגובה הדף (באופן ורטיקלי). כדי למרכז צריך להכיר שתי תכונות:

- **webkit-box-pack** שמיישר את האלמנט לפי הציר שקבענו באמצעות **webkit-box-orient**.
- **webkit-box-align** שמיישר את האלמנט בציר השני. אם קבענו שה-**box-orient** יהיה **horizontal**, אז קובע את היישור בציר הורטיקלי, ולהפך.

יישור למרכז לרוחב הדף

את התכונה הזו מוסיפים לאלמנט שעוטף את העמודות, **wrapper**. הקוד יראה כך:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
    #wrapper {
      width: 100%;
      border: 1px solid black;
      display: -webkit-box;
      -webkit-box-orient: horizontal;
      -webkit-box-pack: center;
    }

    .column {
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
    }
  </style>
</head>
<body>
  <div id="wrapper">
    <p class="column">עמודה ראשונה</p>
    <p class="column">עמודה שניה</p>
    <p class="column">עמודה שלישית</p>
  </div>
</body>
</html>
```

יישור למרכז אורך הדף

webkit-box-pack יכול למרכז את האלמנטים אם הוא מקבל את הערך **center** בנוסף, אם יקבל את הערך **end** ידביק את האלמנטים לצד השמאלי של האלמנט האב, ואם יקבל את הערך **start** ידביק את האלמנטים לצד הימני.

כמובן, שאם היינו ממקמים את האלמנטים אחד מעל לשני ולא אחד לצד השני, start היה מדביק את האלמנטים במעלה הדף, ו-end היה מדביק את האלמנטים בתחתית הדף.

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
    #wrapper {
      border: 1px solid black;
      height: 250px;
      display: -webkit-box;
      -webkit-box-orient: horizontal;
      -webkit-box-pack: center;
      -webkit-box-align: center;
    }

    .column {
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
      height: 150px;
    }
  </style>
</head>
<body>
  <div id="wrapper">
    <p class="column">עמודה ראשונה</p>
    <p class="column">עמודה שניה</p>
    <p class="column">עמודה שלישית</p>
  </div>
</body>
</html>
```

יכולות-**webkit-box-align** :

- יכול למקם אלמנטים במרכז הדף כשמשתמשים בערך **center**
- הוא גם יכול, למקם אלמנטים במעלה הדף כשמשתמשים בערך **start**
- בתחתית הדף כשמשתמשים בערך **end**
- אפשר גם להשתמש בערך **stretch** שיימתח את האלמנט לכל גובהו של הדף

באותה מידה, אם האלמנטים היו ממוקמים אחד מעל לשני אז **center** היה ממקם את האלמנטים במרכז לרוחב הדף, **start** היה ממקם את האלמנטים בימין, ו-**end** בשמאל.

מתיחת האלמנטים בלי חישובים באמצעות CSS3

נניח שאנחנו רוצים שהאלמנט המרכזי ייקח את כל השטח הפנוי לכל אורך הדף, וישאיר לעמודה הראשונה והשלישית את הצדדים. לשם כך, לא צריך חישובים מסובכים כי ב-CSS3 עומדת לרשותנו התכונה **webkit-box-flex**.

לדוגמה, אם ברצוננו למתוח רק את העמודה השנייה, נשנה את קוד ה-HTML וניתן לעמודה **class** ייחודי.

בכדי למתוח את העמודה השנייה, נוסיף לקוד ה-CSS את השורות עבור **column_stretch** נשים לב במיוחד לשורה

webkit-box-flex:1

שמורה לדפדפן למתוח את האלמנט לכל האורך, כך שלא יישאר "שטח לבן" מלבד שוליים. נסתכל על התוצאה. אפשר לראות שהאלמנט המרכזי עכשיו מתוח וממלא את כל החלל שבין שני האלמנטים שמצדדיו.

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
    #wrapper {
      border: 1px solid black;
      height: 250px;
      display: -webkit-box;
      -webkit-box-orient: horizontal;
      -webkit-box-pack: center;
      -webkit-box-align: stretch;
    }

    .column {
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
    }

    .column_stretch {
      -webkit-box-flex: 1;
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
    }
  </style>
</head>
<body>

  <div id="wrapper">
    <p class="column">עמודה ראשונה</p>
```

```

    <p class="column_stretch">עמודה שנייה</p>
    <p class="column">עמודה שלישית</p>
  </div>

```

```

</body>
</html>

```

עכשיו נאמר, שאנחנו רוצים שהאלמנטים יהיו בעלי רוחב שווה. את המטרה הזו ניתן להשיג ללא קושי באמצעות ערך **box-flex** של 1 לכל האלמנטים. הדפדפן יידע לתת לכל האלמנטים בדיוק את אותו רוחב, ללא תלות ברוחב המסך. כך ייראה הקוד:

```

<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
    #wrapper {
      border: 1px solid black;
      height: 250px;
      display: -webkit-box;
      -webkit-box-orient: horizontal;
      -webkit-box-pack: center;
      -webkit-box-align: stretch;
    }

    .column {
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
      -webkit-box-flex: 1;
    }

    .column_stretch {
      -webkit-box-flex: 1;
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
    }
  </style>
</head>
<body>

  <div id="wrapper">
    <p class="column">עמודה ראשונה</p>
    <p class="column_stretch">עמודה שנייה</p>
    <p class="column">עמודה שלישית</p>
  </div>

```

```
</body>
</html>
```

אם נרצה שהעמודה האמצעית תהיה בדיוק רחבה פי-2.8 מכל עמודה אחרת. גם למקרה זה ניתן להשתמש ב-**box-flex**. פשוט נקפיד שהעמודה האמצעית תקבל ערך **box-flex** של 2.8, והעמודות הצדדיות יקבלו ערך **box-flex** של 1. זה בדיוק מה שעושה הקוד הבא:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
    #wrapper {
      border: 1px solid black;
      height: 250px;
      display: -webkit-box;
      -webkit-box-orient: horizontal;
      -webkit-box-pack: center;
      -webkit-box-align: stretch;
    }

    .column {
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
      -webkit-box-flex: 1;
    }

    .column_stretch {
      -webkit-box-flex: 2.8;
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
    }
  </style>
</head>
<body>

  <div id="wrapper">
    <p class="column">עמודה ראשונה</p>
    <p class="column_stretch">עמודה שנייה</p>
    <p class="column">עמודה שלישית</p>
  </div>

</body>
```

</html>

כך זה נראה כשנריץ על הדפדפן. אפשר לראות שרוחבו של האלמנט האמצעי תמיד יהיה פי-2.8 מהאלמנטים הצדדיים ללא תלות ברוחב המסך.

נשים לב ש **box-flex** יכול לקבל כל ערך שלם או בשברים עשרוניים. ערך מיוחד ש-**box-flex** יכול לקבל הוא ערך של אפס, ובמקרה כזה אנחנו צריכים לתת לאלמנט רוחב קבוע באמצעות התכונה width. הקוד יראה כך:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
    #wrapper {
      border: 1px solid black;
      height: 250px;
      display: -webkit-box;
      -webkit-box-orient: horizontal;
      -webkit-box-pack: center;
      -webkit-box-align: stretch;
    }

    .column {
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
      -webkit-box-flex: 0;
      width: 200px;
    }
  </style>
</head>
<body>

  <div id="wrapper">
    <p class="column">עמודה ראשונה</p>
    <p class="column">עמודה שנייה</p>
    <p class="column">עמודה שלישית</p>
  </div>

</body>
</html>
```


שינוי סדר האלמנטים בעזרת CSS3

הודות ל -CSS3 נוכל לשנות את סדר העמודות:

- קודם ניתן לכל עמודה שם מחלקה נפרד בקובץ `html`

• אח"כ נשתמש בתכונה **ordinal-group** שתגדיר את המיקום של כל אלמנט משמאל לימין. אלמנט שיקבל ערך **ordinal-group** של 1 יהיה הכי שמאלי, זה שיקבל ערך 2 יהיה ימני יותר, וכיו"ב.

הקוד יראה כך:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
  <meta charset="utf-8">
  <style>
    #wrapper {
      border: 1px solid black;
      height: 250px;
      display: -webkit-box;
      -webkit-box-orient: horizontal;
      -webkit-box-pack: center;
      -webkit-box-align: stretch;
    }

    .column_right {
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
      -webkit-box-flex: 1;
      width: 200px;
      -webkit-box-ordinal-group: 2;
    }

    .column_left {
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
      -webkit-box-flex: 1;
      width: 200px;
      -webkit-box-ordinal-group: 1;
    }

    .column_stretch {
      -webkit-box-flex: 2.8;
      border: 1px solid black;
      background-color: yellow;
      padding: 10px;
      margin: 7px;
    }
  </style>
</head>
</html>
```

```

        -webkit-box-ordinal-group: 3;
    }
</style>
</head>
<body>

    <div id="wrapper">
        <p class="column_right">עמודה ראשונה</p>
        <p class="column_stretch">עמודה שנייה</p>
        <p class="column_left">עמודה שלישית</p>
    </div>

</body>
</html>

```

תכונה מאוד שימושית לאתרים דו-לשוניים שבהם עוברים בין השפה העברית שכתובה מימין לשמאל לשפה שכתובה משמאל לימין דוגמת אנגלית. במצב כזה, אנחנו יכולים להדרש להפוך את סדר העמודות. לשם כך, נוכל להשתמש בתכונה **webkit-box-direction** עם ערך של **reverse** או עם הערך ברירת המחדל **normal**

```

<!DOCTYPE html>
<html>
<head>
    <title></title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <style>
        #wrapper {
            border: 1px solid black;
            height: 250px;
            display: -webkit-box;
            -webkit-box-orient: horizontal;
            -webkit-box-pack: center;
            -webkit-box-align: stretch;
            -webkit-box-direction: reverse;
        }

        .column_right {
            border: 1px solid black;
            background-color: yellow;
            padding: 10px;
            margin: 7px;
            -webkit-box-flex: 1;
            width: 200px;
            -webkit-box-ordinal-group: 2;
        }

        .column_left {
            border: 1px solid black;
            background-color: yellow;
            padding: 10px;

```

```

        margin: 7px;
        -webkit-box-flex: 1;
        width: 200px;
        -webkit-box-ordinal-group: 1;
    }

    .column_stretch {
        -webkit-box-flex: 2.8;
        border: 1px solid black;
        background-color: yellow;
        padding: 10px;
        margin: 7px;
        -webkit-box-ordinal-group: 3;
    }
}
</style>
</head>
<body>

<div id="wrapper">
    <p class="column_right">עמודה ראשונה</p>
    <p class="column_stretch">עמודה שנייה</p>
    <p class="column_left">עמודה שלישית</p>
</div>

</body>
</html>

```

סיכום תכונות מודל התיבה הגמישה

משמעות	ערכים	Chrome
מאתחל	אין	display: -webkit-box
מסדר אלמנטים לגובה או לרוחב	horizontal vertical	webkit-box-orient-
0 רוחב קבוע, 1 רוחב גמיש, גדול מ-1 - רחב פי כמה מאלמנט שמוגדר כ-1	1, 0 או כל ערך גבוה מ-1 כולל עשרוני	webkit-box-flex-
ממקם את האלמנט ביחס לציר שנקבע על ידי box-orient	start end center	webkit-box-pack-

ממקם את האלמנט בציר ההפוך	start end center stretch	webkit-box-align-
סידור האלמנטים הילדים מימין לשמאל או מלמעלה	normal reverse	webkit-box-direction-
מיקום האלמנט	מספר שלם	webkit-box-ordinal-group-