

## מדריך CSS

### תוכן עניינים

2	מבוא
4	הטמעת עיצוב בתגיות HTML
7	Margin & padding
9	מיקומים - Display, Float, Positions
12	תכונות חדשות בCSS3
14	אנימציות
16	רספונסיביות - Media Query
18	סלקטורים
20	Bootstrap
22	FlexBox
26	SASS

## מבוא

לאחר שיש לכם ידע בשפת HTML וכיצד נראה שלד של אתר כלשהו עם תגיות מובנות מראש, ניתן לקחת את אותן תגיות מובנות מראש  
body , div , header , footer , h1, span, p , input ועוד...  
ולשנות להן את תכונות ברירת המחדל שהגדירו המתכנתים של הHTML.

לצורך העניין, תכונות ברירת המחדל של p זה: צבע שחור, פונט משעמם, גודל מסוים, מרחק ממי שמעל הפסקה ומרחק ממי שמתחת לפסקה. למעשה דרך ה CSS ניתן להשתלט על התכונות הללו ולשנות אותן.

כל שצריך לעשות הוא להוסיף בתוך כל תגית פותחת את המילה style. מיד לאחריה לרשום שווה (=) ואז בתוך גרשיים את כלל התכונות ואת הערכים שלהן (למשל תכונה: צבע, ערך: כחול) בהפרדה של נקודה פסיק בין עיצוב לעיצוב. למשל, שם התכונה של צבע הטקסט הוא color וצבע הרקע הוא background-color. אם נרצה שצבע הטקסט של כל הכיתוב ב div שלנו יהיה אדום, נצטרך לרשום זאת כך. color:red -

זה הזמן שלכם - נסו ליצור קובץ HTML עם head, body ובתוך ה body לשים div כמו בדוגמא הבאה:

```
<div style='color:red;background-color:pink'></div>
```

בדיוק כמו בדוגמא הקודמת, כך תוכלו בעצמיכם ליצור אלמנטים שונים בbody שלכם ולשנות להם תכונות שונות. עוד מעט מן התכונות הבסיסיות בcss:

- `font-size:15px` - גודל הפונט. תמיד אחרי הנקודותיים נציין את הגודל בצירוף px.
- `font-weight:bold` - טקסט בולט בדף
- `text-decoration:underline` - הוספת קו מתחת לטקסט
- `text-align:center` - יישור הטקסט שבתוך div למרכז.
- `height:50px` - הוספת גובה לdiv
- `width:100%` - הוספת רוחב. ניתן במקום באחוזים.

שימו לב, במקום כל הערכים שהזנתי מראש (למשל ה100% או הצבע האדום) אתם יכולים לשנות את הערכים כרצונכם, כל עוד הcss מכיר את התכונה ואת הערך. למשל, במקום red תכתבו blue או yellow. במקום 100% אפשר גם 50% וכך הלאה... כנ"ל לגבי center, אפשר גם left/right. או underline יש גם קו חותך אמצע וקו עליון על טקסט... פשוט לחקור את הנושא עוד ועוד ברשת... לסיום השיעור, נסו לאתגר את עצמכם. מלבד שיפוץ האתר שבניתם בשלב הHTML - תרגיל לאתגר הפעם: צרו ריבוע בדף שלכם. יש לריבוע גובה, רוחב וצבע רקע צהוב. בהצלחה.



## הטמעת עיצוב בתגיות HTML

בשיעור הקודם הצגנו לפניכם רשימת תכונות בסיסיות ב CSS אשר ניתן להטמיע אותן על תגיות HTML הקיימות מראש.

למעשה, כל תגית HTML מכילה כבר תכונות משלה - למשל, תגית `h1` מכילה גובה, טקסט בולט, בצבע שחור, בפונט מסוים, יש לה מרחק מתגיות אחרות וכדומה... אנחנו כעת נבצע 'דריסה' של התכונות הקיימות ונשנה אותן לתכונות שנרצה אנו...

---

### כעת נלמד איך להטמיע את התכונות העיצוביות

לפנינו 3 דרכים - האחת, כמו שלמדנו בשיעור הקודם. נוכל לייצר בתוך התגית אזור לתכונה בשם `style`, לאחריה שווה, לאחריו גרשיים ובתוך הגרשיים להזין את התכונות שנרצה. בין תכונה לתכונה נקפיד להפריד בנקודה פסיק; שהרי רק כך הרובוט שקורא את האתר שלנו בדפדפן, יידע להבחין בין תכונה לתכונה. כמו שב HTML תמיד אנו סוגרים את התגית.

השיטה השניה היא לייצר אזור בתוך ה `head` עם תגית `<style></style>`.

```
<style>
div {
color: blue;
text-align:center;
text-decoration:underline
font-family:Tahoma;
}
</style>
```

ולסיום, השיטה הנכונה והפופולארית ביותר כיום הינה שיטה של ייבוא קובץ css חיצוני. בתוך קובץ זה נוכל ליצור את כל קטעי העיצוב השונים, הקריאות ל dom של HTML שלנו ולשנותו בהתאם.

ראשית, בתוכנת notepad צרו קובץ נוסף ובתפריט העליון תחת שפות הגדירו כשפת css. בתוכה אין צורך יותר לקרוא לתגית style, אלא פשוט להתחיל לכתוב את איזורי התגיות בסוגריים מסולסלים כמו שביצענו בשיטה הקודמת.

השיטה השלישית היא הנכונה ביותר והפופולארית ביותר. נרצה שכל קובץ באתר שלנו יציג שפה אחידה מבלי קשקושים או ערבובים בין שפות, על אף שהדבר אפשרי. ראינו את שיטות 1 ו 2 אשר מציגות את הכתיבה בתוך דף HTML, מה שעלול ליצור קוד מסורבל, ארוך ומלא בתגיות HTML שזורות בעיצוב CSS. השיטה השלישית מציגה שורה אחת אותה תשלבו בחלק של ה head ב HTML ומטרת שורת קוד זו היא לייבא את קובץ ה CSS שלנו שיהיה חיצוני. הפעם בקובץ החיצוני לא נצטרך להזין תגית style פותחת ו style סוגרת, אלא פשוט להזין את העיצוב שנרצה על התגיות השונות.



## מדריך CSS מאתר "עולם ההייטק" המקורי ©

התגית המדוברת היא תגית link, זו תגית ריקה שנפתחת ונסגרת באותה עת והיא מכילה שתי תכונות. האחת rel כלומר מה הקשר של היבוא - במקרה זה stylesheet (קיצור של css) וכן תכונה נוספת של href שבתוכה נזין את את src עצמה.

```
<link href="css/style.css" rel="stylesheet">
```

---

בימינו ישנן ספריות המתלבשות על css ומסוגלות לבצע פונקציות מתקדמות, לשלב תכנות על גבי עיצוב ה css ועוד ועוד... אולם, דבר דבר - תחילה נכיר את יסודות שפת העיצוב של התגיות ובהמשך נתקדם גם לביצוע בדיקות שונות, תכנות מתקדם על העיצוב וכדומה. כעת - נייצר תגית ב html ונעצב אותה עם תכונות בסיסיות ב css. הצטרפו גם לשיעורים הבאים.

## Margin & padding

אחד הנושאים המורכבים ביותר ללומדים הוא נושא זה. פקודות הmargin והpadding. כאן בעולם ההייטק, אלמד אתכם זאת בפשטות ובקלות ללא להסתברך.

### ראשית, נתחיל מmargin:

כאשר אנו רוצים להרחיק את האלמנט שלנו מאלמנטים אחרים בדף נשתמש בmargin. דמיינו, שאני עומד בכיתה ומתרחק מהכיסא. הכיסא נשאר במקום אך אני מתרחק. אותו דבר גם כאן. צרו ריבוע עם תכונות של גובה, רוחב וצבע רקע. כמו בדוגמא:

```
<div style='height:25px;width:25px;background-color:yellow'>  
</div> יש לנו ריבוע ואנחנו יכולים כעת להתחיל לשחק איתו. אם נמשיך  
להזין תכונות בתוך הגרשיים בstyle ונרשום שם את התכונות הבאות, הריבוע  
יתרחק כמספר שנדון:
```



- margin-top:15px
- margin-right:15px
- margin-bottom:15px
- margin-left:15px

ניתן גם לכתוב זאת כתכונה אחת כאשר יש לה 4 ערכים כסיבוב השעון:

```
margin:15px, 0px, 5px, 10px
```

כלומר, יהיה מרחק של 15 פיקסלים מלמעלה, 0 מימין, 5 מלמטה, 10 משמאל (כסיבוב השעון).



## מדריך CSS מאתר "עולם ההייטק" המקורי ©

בתכונת הpadding, אנו גורמים div שלנו להיות שמן יותר או רזה יותר. זה יהיה כמו לכתוב width או height במובן מסוים. שימו לב:



- `padding-top:15px`
- `padding-right:15px`
- `padding-bottom:15px`
- `padding-left:15px`

או לחלופין: `padding:15px, 0px, 5px, 10px`

כעת, בדרך הpadding הdiv שלנו יקבל מרחק מלמעלה ב15 בכך שהוא השמין ולא התרחק כמו בmargin. כנ"ל לגבי שאר התכונות. נסו בעצמיכם.

זהו ההבדל בין padding לmargin. margin מרחיק את האלמנט, padding משמין את האלמנט ומנפח אותו.

בשילוב עם borders ותוכן, מודל זה נקרא גם box model.



## מיקומים - Display, Float, Positions

בשיעור זה אלמד על שלושה אלמנטים שונים בcss:

נתחיל מdisplay:

לdisplay יש מספר ערכים שניתן להזין ולכל אחד שימוש אחר...

למשל, הdefault של כל אלמנט הוא `display:block` כלומר - תהיה גלוי. תופיע במסמך.

אבל אפשר גם להעלים אובייקט כלשהו. כלומר, ליצור אותו, לכתוב בHTML הכל אבל שהוא לא

יופיע. נכתוב זאת כך: `display:none`

יש גם את `display:inline-block` שמשמעותו היא לגרום לdiv שלנו, שאליו ייחסנו את התכונה

הזו, לעמוד ליד אלמנטים אחרים. מפני שמצב ברירת המחדל של הdiv הוא שהוא עומד איתן

לבדו. ניתן לגרום לשים לידו אלמנטים נוספים.

נמשיך לנושא הfloat:

גם float גורם לאלמנטים 'לצוף' ולהיות אחד ליד השני. אמנם, תוך כדי תכנות, יש הבדלים בין inline-block לבין float. נסו בעצמיכם:

```
float:right
```

```
float:left
```

על מנת לבטל את הfloat הקיים, ניתן להשתמש אם תרצו ב `clear:both`

לסיום, נושא הpositions:

יש בcss אפשרות לגרום לאלמנטים 'לרחף' על גבי הדף ללא תלות או קשר לאלמנטים אחרים. ניתן להגדיר לאלמנט את המיקום היחסי שלו בדף. צריך לזכור, כי כל האתרים בדפדפנים השונים מתחילים במצב הdefault בנקודת 0,0. כמו מערכת צירים. ציר הy (גובה) הוא 0, ציר הx (רוחב) הוא 0. גם כאן, בהנחה שאין לכם בbody דבר ותצרו div בצורת ריבוע הוא יהיה אוטומטית בנקודת ה0,0 בצד שמאל למעלה. ניתן לשנות זאת! ראשית, עליכם להגדיר לdiv שהוא `position:absolute` ולאחר מכן את המיקום היחסי שלו בדף קרי הדוגמא:

- `top:15px`
- `bottom:15px`
- `right:15px`
- `left:15px`

במידה ונרצה שהאלמנט שלנו יהיה מתחת לטקסט ולא מעליו, כמו שכבה נוספת שכזו - הכירו את `z-index`.

זוהי תכונה אשר משמשת לשכבות שונות על הדף. מצב הברירת מחדל הוא 0.

מלבד `absolute` יש גם את `fixed`. הוא נפוץ ביותר ומופיע כמעט בכל אתר! `z-index: -1`

כאשר נגדיר לאלמנט מסוים את אותן התכונות כמו לעיל, אך עם השינוי הקל ונהפוך אותו ל `fixed` נקבל מצב בו האלמנט 'צף'/'מרחף' כל הזמן באותה נקודה באתר לא משנה איפה אנחנו בדף. נסו בעצמיכם!

## תכונות חדשות בCSS3

השדרוג החדש לCSS והגרסא של CSS מס' 3 מציגה בפנינו המון תוספים חדשים ותכונות מיוחדות. בואו נצא לדרך עם חלק נכבד מהן שמשדרגות את האתר שלנו מבחינת העיצוב:

- `border-radius: 25px;` במקום ריבוע, תוכלו לעגל את פינות div. ניתן לפרק ל4 ערכים לפי סיבוב השעון.
- `opacity: 0.6;` הערכים נעים על סקלה בין 0 ל1. מציג את הצבע/תמונה במצב של שקיפות יחסית.
- `text-shadow: 2px 2px 5px red;` הוספת צל לטקסט. מכיל כ4 ערכים. הראשון זה המרחק מהtop, השני המרחק משמאל, השלישי חדות הצל והאחרון צבע הצל.
- `box-shadow: 10px 10px 5px grey;` אותו דבר כמו צל לטקסט, הפעם צל לdiv עצמו.

הערה חשובה: מפני שהרבה דפדפנים לא תומכים בCSS3 ניתן לבצע כתיבת קוד למצב ברירת מחדל. כלומר, מה קורה במצב בו הדפדפן לא תומך בgradient? ניצור גם background-color רגיל.

```
background: red;  
background: linear-gradient(red, yellow);
```

לאחר שאתם יודעים כבר מהו CSS וכיצד כותבים ומעצבים, לא תהיה בעיה עבורכם לחקור ברשת המלאה במידע ולהטמיע את התכונה לאתרכם.

מדריך CSS מאתר "עולם ההיטק" המקורי ©



## אנימציות

אנימציות הפעם, נלמד על אנימציות ושינויים דינאמיים בCSS.

ראשית, נצטרך לכתוב הכל בcss חיצוני.

css חיצוני הכוונה היא שניתן לפתוח בnotepad++ קובץ חדש ובסרגל העליון תחת 'שפה' לבחור ב'css' או 'גליונות סגנון מדורגים'. כעת, תוכלו לכתוב שם את כל העיצוב שלכם בצורה מסודרת להפליא. כל שעליכם לעשות הוא להגדיר 'איזור' לכל אלמנט. למשל, אם נרצה שיהיה לנו צבע רקע כחול, צבע טקסט אדום וקו מתחת לטקסט בכל תגית h1 נבצע זאת בcss החיצוני כך:

```
h1 {  
background-color:blue;  
color:red;  
text-decoration:underline;  
}
```

לסיום, כל שנצטרך לעשות הוא לתת הפנייה לcss החיצוני שלנו בתוך הhead. בדרך הבאה:

```
<link href='style.css' rel='stylesheet'>
```

כאשר נרצה לבצע מצב בו האלמנט שלנו יהפוך מצבע צהוב לאדום בצורה מהירה ודינאמית אל נגד עיני המשתמש שלנו בצורת אנימציה, נוכל לבצע זאת כך: כעת, עלינו להגדיר בcss החיצוני איזור חדש לחלוטין:

```
@keyframes tal {  
  from {background-color: red;}  
  to {background-color: yellow;}  
}
```

כפי שניתן לראות בסוגריים המסולסלים יש את ה from והתכונות של מצב ברירת המחדל ואת המצב העתידי אליו אנו רוצים להגיע ב to.

נצטרך גם להגדיר מראש @keyframes tal שהה tal יכול להשתנות (שם האנימציה למעשה, לבחירתכם) אולם ה keyframes@ חייב להיות כמו שהוא (קריאה לאנימציה).

בתוך ה div עצמו תקראו לאנימציה ותוכלו להוסיף עוד תכונות כמו כמה זמן היא תימשך? האם לנצח? וכדומה.

```
animation-name: tal;  
animation-duration: 4s;  
animation-delay: 2s;  
animation-iteration-count: infinite;
```



## רספונסיביות - Media Query

בשיעור זה נלמד כיצד להתאים את האתר לגדלים שונים של מכשירים שונים. לדוגמא, האתר שלנו בעולם ההייטק נראה בצורה מסוימת במחשב אך מותאם, ככל שניתן, גם לצפייה נקייה וברורה לכל גולש דרך הסמארטפון שלו. אין לנו שני אתרים או שני דומיינים - אלא - יש לנו אתר אחד והוא מחולק ל 2 קבצי css שונים.

לאחר שעיצבתם אתר משלכם המותאם למחשב דרך css חיצוני, פיתחו עוד css חיצוני חדש וצרו בו את הקוד הבא:

```
@media screen and (min-width: 480px) {  
  body {  
    background-color: yellow;  
  }  
}
```

על פי הקוד הנ"ל, כל פעם שהמסך שלנו יהיה פחות מ 480 פיקסלים, צבע הרקע ישתנה לצהוב. נסו בעצמיכם: צרו אתר עם `div h1`. כאשר האתר בגודל קטן מ 480, שנו את הצבעים של כל האלמנטים בדף (גם של הרקע). תוכלו להקטין את המסך ולהיווכח בעצמכם שאכן יש שינויים בדפדפן.

למכשירים שונים יש גדלים שונים ולכן מומלץ לחפש בgoogle את הגודל הרלוונטי שאתם רוצים לאתרכם. צפו בסרטון ההסברה והתאימו את האתר שלכם לגדלים שונים.

מדריך CSS מאתר "עולם ההיטק" המקורי ©



## סלקטורים

ניתן לדבר עם אלמנטי html שונים דרך css שלנו. ראינו מספר דרכים לכך. ניתן דרך התגית עצמה, ניתן דרך css חיצוני ואפשר דרך תגית style בתוך הhead. כאשר נרצה לפנות לdiv ספציפי למשל ולא לכלל הdiv שיש לנו בדף, נצטרך לתת לו שם ייחודי. הדבר נקרא id. אנו פונים לid בcss דרך סולמית #.

```
<div id='tal'></div>
```

```
<style>  
#tal {  
color: blue;  
}  
</style>
```

כאשר נרצה לפנות לחבורת div ספציפיים למשל ולא לכלל הdiv שיש לנו בדף, נצטרך לתת לקבוצה הזו שם ייחודי. הדבר נקרא class. אנו פונים אל הclass בcss דרך נקודה.

```
<div class='tal'></div>  
<div class='tal'></div>  
<div class='tal'></div>
```

```
<style>  
.tal {  
color: blue;
```

```
}  
</style>
```

נסו בעצמיכם!

עוד מן עולם הסלקטורים והאפשרויות לקרוא לאלמנטים שונים בcss לפניכם:

- \* - בחירת כלל האלמנטים
- `div > h1` - בחירת רק תגיות הh1 שהן הילדים של הdiv
- `div:hover` - במעבר עכבר על div תבצע משהו
- `div:eq(0)` - בתוך הסוגריים נזין מספר (הספירה מתחילה מ0) ונקרא לאלמנט הdiv - במקום הראשון, השני וכו'

## Bootstrap

ישנה חבורת מתכנתים שכתבו עבורינו ספריות שלמות של CSS המכילות classes שונים, קריאות ל ID בשמות ספציפיים וגם לתגיות HTML רגילות כמו `input`, `button`, וכדומה... בתוך קובץ CSS החיצוני שהם יצרו, הם שינו את מצבי ברירת המחדל לתכונות עיצוביות מרהיבות משלהם ואנו כמתכנתים משתמשים בכך.

כל שעלינו לעשות הוא להזין את שורת CSS החיצוני (ניתן גם להוריד את קובץ CSS של bootstrap למחשב) בתוך `Head` שלנו ולהתחיל לחקור ולבדוק איך לעצב על פי ההגדרות שלהם.

```
<link rel='stylesheet'  
href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css'>
```

למשל אחד הכלים השימושיים ביותר הוא class `<div class='container'></div>` אשר גורם לאלמנטים שלנו להיות במרכז הדף. גם במצב של גודל מסך גדול או גודל מסך קטן - האלמנט עדיין יישאר באמצע. נסו בעצמיכם.

---

עוד מן האפשרויות השכיחות יותר על bootstrap היא גם שימוש בכפתורים מעוצבים אשר מכילים כבר סימנים מוטבעים מראש בתוך `button` או `input` (למשל, `input` שמכיל חיפוש בדף). בנוסף לאלה, יש גם אפשרות לחלק את הדף ל 12 חלקים כמעין טבלה. מאוד שימושי ונוח לסדר קצת אלמנטים זה לצד זה. כל שצריך לעשות זה להגדיר ב class הרלוונטי כמה נפח הוא תופס מתוך ה 12. למשל, אם זה 6 אז בדיוק חצי... אם 1 אז זה 1/12 מן הדף (איזור קטנטן)

## מדריך CSS מאתר "עולם ההיטק" המקורי ©

וכדומה... זהו למשל קוד אשר מציג class של נפח 4. כל אחד מהשלושה תופס את אותו class והם מגיעים בדיוק ל12.

	col-sm-4.
	col-sm-4.
	col-sm-4.

## FlexBox

בגרסת CSS3 נוסף מצב פריסה ייחודי כתכונה נוספת ב `display` בשם `flex`. מטרת העל של מצב פריסה זה הוא לתת מענה טוב יותר לבעיות חדשות שנוצרו בעידן הדיגיטל שבו אנחנו נמצאים. קרי: התאמה מהירה לרזולוציות מסך שונות, שינוי אוריינטציית מסך, הגדרות ברירת מחדל משתנות לפי דפדפן ועוד...

עולם ה `display: flex` והתכונות שהוא מביא איתו, הינו עולם שלם ומלא בתכונות חדשות ומתקדמות. במדריך זה, אעשה סדר בכל הנוגע לתצוגה הראשונית וסידור של ריבועי div רגילים בזה לצד זה, רק כאשר נקטין או נרחיב את המסך (אפילו ממש את מסך מדריך זה) נראה אותם מסודרים אחרת. בין אם הרוחב שלהם יקטן או בין אם יירדו שורה, בלא כל קשר להתערבות של media query כמו שלמדנו במאמר מקדים למאמר זה. יוצאים לדרך - כך נראה flex בסיסי המכיל 3 divs בתוך שורה סמוכים זה לזה:

```
<div class='flex-container'>  
<div>1</div>  
<div>2</div>  
<div>3</div>  
</div>
```

ובcss לא נשכח להוסיף את תכונת ה `display: flex` החדשה שלנו ישירות לתגית class הראשית של div שלנו - `flex-container`. וכן גם תכונות עיצוביות כמו צבע רקע כחול, טקסט בגודל





## מדריך CSS מאתר "עולם ההייטק" המקורי ©

מסוים וכדומה... מנקודה זו, על מנת לגרום לריבועים שיצרנו ליצור פעולה כלשהי נצטרך להכיר מספר תכונות שיש לflexbox ועלינו יהיה להוסיף את התכונה הרלוונטית לאתר שלנו.

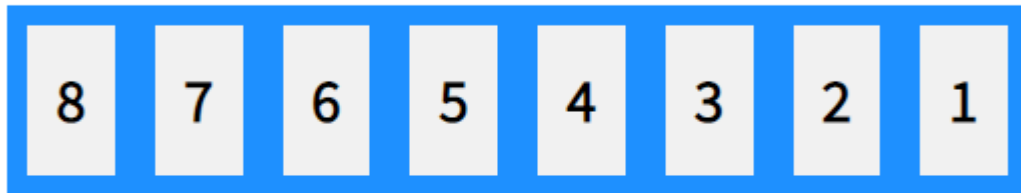
מנקודה זו, על מנת לגרום לריבועים שיצרנו ליצור פעולה כלשהי נצטרך להכיר מספר תכונות שיש לflexbox ועלינו יהיה להוסיף את התכונה הרלוונטית לאתר שלנו.

---

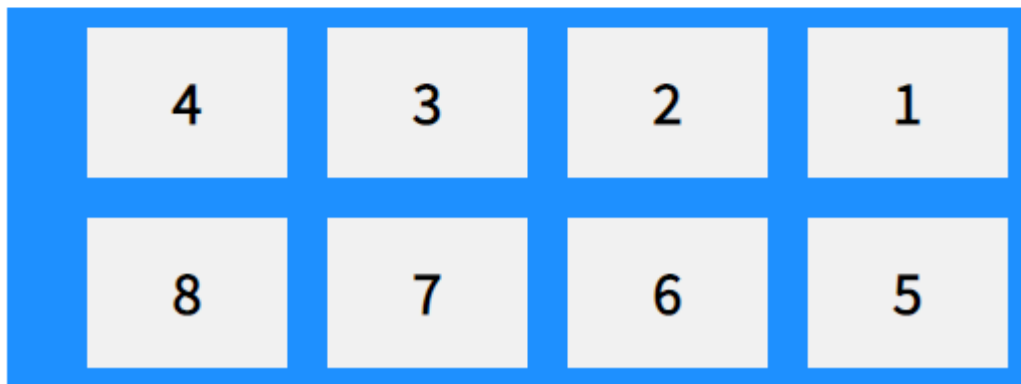
התכונות הקיימות הן למשל:

`flex-wrap` - הכוונה פה היא שהתכונה מדברת על ההתנהגות של האייטמים השונים, כשאינן להם די מקום. הערך במקרה זה `nowrap` מייצג את צמצום ה-divs שלנו והקטנתם. נסו לראות בדוגמא המצורפת מטה. עוד ערכים הם `wrap` למשל שמציין ירידת ה-divs השלישי מטה, במידה ואין לו מקום בקונטיינר שלו.

להלן דוגמא ל `flex-wrap: nowrap`:

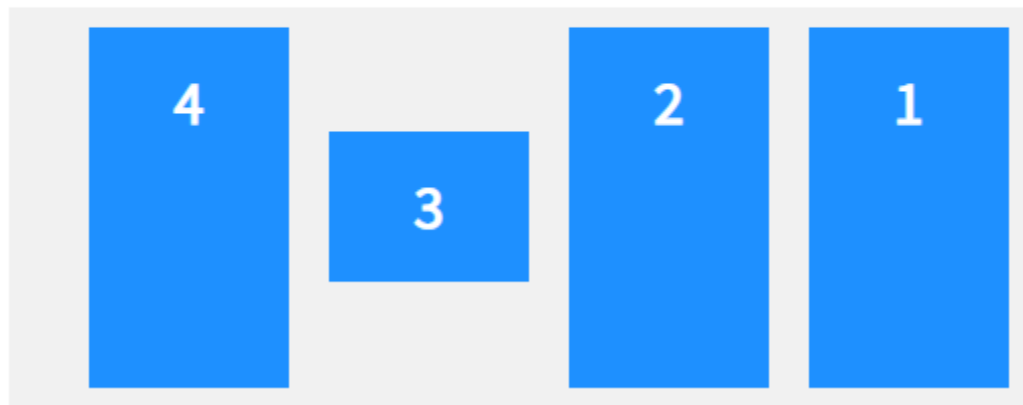


להלן דוגמא ל `flex-wrap: wrap`:



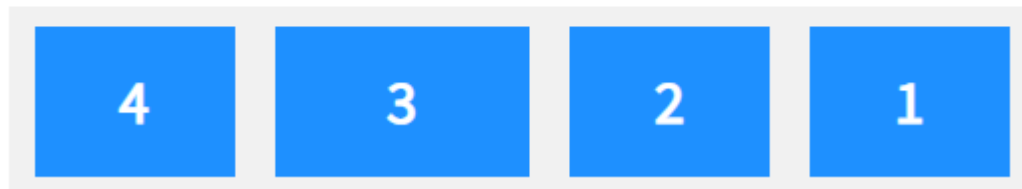
`align-self` - במקרה זה מדובר בתכונה המאפשרת לשנות מיקום של אלמנט ספציפי בתוך הקונטיינר. היא מאפשרת לנו לשבור את הגדרת ברירת המחדל הדיפולטית של `align-items` הקיימת לקונטיינר עבור אלמנט ספציפי בקונטיינר זה.

להלן דוגמא ל `align-self`:



`flex-basis` - מגדירה באחוזים את השטח אותו יתפוס כל אלמנט. למשל ניתן להגדיר כ-25% `div` כלשהו. בדומה לשאר התכונות מעלה עלינו לדבר בcss דרך סלקטורים עם `div` ספציפי בתוך הרשימה - למשל: `div:nth-child(1)`

להלן דוגמא ל `flex-basis: 25%` `div` השלישי:



## SASS

עכשיו כשיש לכם קצת יותר ידע ותרגלתם אתרים מעוצבים, הגיע הזמן להכיר במגבלות של הcss הקלאסי... הוא יכול להגיע לאלפי שורות קוד, עבור כל תגית ותגית צריך לייצר אזור משלה, לא נוכל לחבר בין אזורי עיצוב אם נרצה, לא נוכל להשפיע בין איזור לאיזור, לא נוכל ליצור בדיקות if פשוטות אפילו - למשל, רק אם המשתמש לא הזין טקסט אז תציג `display:block` לdiv שהכנו בצבע אדום עם כיתוב error כלשהו...

בהנחה שהנקודה ברורה, ואנו מבינים כי אנחנו רוצים לשדרג את עולם הcss שנתקל במגבלות רבות, אנו יכולים להישען על ספריות שונות אשר דרכן נוכל ליצור שאילתות שונות, לייצר בדיקות, ליצר השפעה של אזור אחד בcss על אזור אחר וכדומה.

---

מדריך זה הינו מדריך css, ובכל זאת - בשל הפופולאריות שתופסת ספריית הsass אפרט כאן כיצד נוכל להשתמש בה במספר נקודות:

1. ראשית, על מנת לכתוב בשפה זו, עליכם להשתמש במחוללי אונליין של בניית אתרים או להתקין תוכנות התומכות ביצירת server localhost ותומכות בsass. הסיבה לכך היא שהרובוט בדפדפן הרגיל לא קורא בצורה ישירה את סינטקס העיצוב של sass ותוכנות אלו יוצרות את הקומפרסור המתאים כך שמתבצעת דרכן המרה לcss.
2. על מנת ליצור משתנה אחד ראשי (ראו ערך מדריך הלוגיקה) ולאחר מכן פשוט להשתמש בו (מבלי כל פעם למשל לכתוב בעצמינו את אותה השורה. קרי, גודל של טקסט או מרחק

מסוים שחוזר על עצמו) נוכל לייצר משתנה גלובלי עם דולר לפני שם המשתמש, שווה ואז את הערך הרלוונטי.

3. נוכל לאחד בין איזורים. לפתוח אזור של header למשל ובתוכו לפתוח איזור של כל התגיות שבתוכו... במקום שהיינו מייצרים אזור שלם ונפרד לכל תגית בתוך header כמו שתיארנו בשיעור סלקטורים למשל...

---

מוזמן ואף רצוי לעבור לטכניקת sass, אשר גם עליה יש שיפוצרים ופיצ'רים מתקדמים בספריות כמו less למשל... הcss הקלאסי הוא מאוד ידידותי ונוח, אבל גם מאוד מוגבל ומסורבל. עלינו לייעל את הקוד שלנו באמצעות המשתנים, באמצעות איחוד האיזורים, באמצעות בדיקות שונות שניתן לעשות על קובץ sass. בהצלחה