



# CSS מדריך



## הקדמה

### הקדמה

#### **CSS** באנגלית הינה מילת קיצור למושג **Cascading Style Sheets**

שפירושו גיליונות סגנון מדורגים - סגנונות הקובעים כיצד יוצגו האלמנטים של HTML.

סגנונות אלו מאפשרים למתכנת ה-WEB לעצב דפים בנוחות לרוחב היישום כולו. מעצבים מקצועיים משתמשים בקווי עיצוב מוגדרים מראש אותם הם בוחרים לפרויקטים שלהם.

דפוסי עיצוב אלה מלווים את הפרויקט מתחילתו ועד סופו ומקנים לו זהות, אופי וסגנון.

אין מדובר כאן בתוכן כלל. ישנה הפרדה מוחלטת בין קו עיצובי לבין תוכן.

CSS מאפשר למעצב לקבוע כיצד יוצג כל אלמנט בדף HTML, גליונות סגנון הם הכלי האולטימטיבי לאפיון, עיצוב ועימוד תכני HTML.

## לשם מה צריך גיליונות סגנון



תגיות HTML נוצרו במקור כדי להגדיר את הצגת התוכן במסמך, הם אמורים להגיד לגבי תוכן מסוים אם הוא מהווה פסקה, טבלה, רשימה, קישור וכו'. (תגיות P, Li, table)

עימוד המסמך היה צריך להיעשות על ידי הדפדפן, עם הזמן הוסיפו תכונות עיצוב עבור תגיות ה HTML כגון תגית <font> והתכונה שלה עבור צבע (color="red"), או לחילופין תגית היוצרת קו מפריד אופקי </hr> והגדרת הרוחב שלה (width="50%"), עד שבבניית אתרי אינטרנט לא ניתן היה למצוא את קו ההפרדה שצריך להיות קיים בין עיצוב המסמך, לבין התכנים, ולבין התכונות. הכל הפך למסה אחת שקשה למצוא בה את הידיים והרגליים. עיצוב דפי WEB הפך לסיוט של ממש כדי לפתור בעיה זו החליטו חברי ועדת W3C (הועדה ללא מטרת רווח המחליטה על תקנים של HTML), ליצור סגנונות עיצוב. כמו כן, יצרה הועדה את התקן החדש - HTML 4.0.

גיליונות הסגנון נקראים "מדורגים" (באנגלית: Cascading) משום שיש להם היררכיה: אם יש הגדרות שונות עבור אותו אלמנט בדף, הגדרות CSS הנמצאות בבולק <style> בחלק ה- <head> של דף ה-HTML גוברות על הגדרות מקבילות הנמצאות בדף CSS חיצוני, והגדרות CSS הנמצאות בתוך תגית בדף (כערך של התכונה style) גוברות על הגדרות בבולק <style>.

## מה צריך לדעת כדי ללמוד CSS

דרוש ידע בשפת HTML אשר משמשת להוספת תוכן עבור אתר אינטרנט.

תוכן הכוונה ל:

- פסקאות

- כותרות

- קישורים

- רשימות

- טבלאות

- תמונות

- וכו'

+

-

## באמצעות אילו כלים כותבים CSS

כל עורך טקסט פשוט, כדוגמת notepad, מספיק כדי למלא את המשימה.

מומלץ יותר להשתמש בעורך מתקדם יותר אשר מאפשר מעקב אחר טעויות הקלדה ומאפשר השלמה אוטומטית של פקודות קיימות כגון: web developer או visual studio

כמובן שקיימים עורכים נוספים חინמיים שניתן להוריד באינטרנט אשר יספקו את אותן התוצאות.  
בשיטוט קל באינטרנט והקלדת מילות חיפוש כגון : HTML Editor או CSS editor תוכלו למצוא מגוון תוכנות נחמדות לשימוש.

## יתרונות השיטה

הפרדה זו יוצרת מספר יתרונות:

### עיצוב גלובלי:

בראש ובראשונה יצירת תבנית גלובלית לכל דפי האתר.  
שינוי דף ה-CSS ישתקף בכל דפי האתר, ולא יצריך התערבות פרטנית בכל דף ודף.

### קוד נקי יותר:

הורדת המורכבות והסיבוכיות ביצירת דפי HTML, באמצעות הוצאת הוראות העיצוב מתוכם.

+

הפרדת רשויות:

-

הפרדה נוחה בין אנשי המקצוע השונים שעובדים ביצירת האתר. העיצוב והגרפיקה נעשים על ידי אנשי העיצוב בקבצים נפרדים, והתכנים נכתבים על ידי אנשי התוכן בקוד נקי מהגדרות עיצוב.

התאמה אישית:

ההפרדה בין העיצוב לתכנים מאפשרת הצגה מותאמת למשתמש, שבה כל משתמש יכול לבחור באיזה עיצוב, מתוך עיצובים אחדים המוצעים לו, ברצונו לצפות באתר.

התאמה לפלטפורמות שונות:

שימוש ב-CSS מאפשר יצירת גרסאות להדפסה, גרסאות להצגה על מדיומים אחרים (כמו מחשבי כף יד, טלפונים סלולאריים טלויזיה וכדומה). CSS תומכת גם ביכולות קוליות ובכתב ברייל.

אופן קביעת הסיגנון

ארבעה אופני קביעת הסיגנון

ניתן לקבוע סגנון בארבעה אופנים:

1. ברירת מחדל - לכל דפדפן קיימות הגדרות עיצוב ברירת מחדל מבחינת גודל וסגנון כתב ועוד. ב-CSS מגדירים בדרך כלל רק את התכונות שאנו רוצים לשנות ואת כל שאר התכונות משאירים ללא שינוי.
2. Inline - סוג זה יוסבר בהמשך הפרק
3. Internal - סוג זה יוסבר בהמשך הפרק

#### 4. External - סוג זה יוסבר בהמשך הפרק



Inline



את הגדרת העיצוב ניצור בתגית פתיחה במידה ומדובר על תגית

מכולה) או בתגית ריקה בעזרת התכונה **style**

למשל לתגית `<p>` המסוימת הזאת אנו נקבע שהרקע יהיה צהוב:

```
<p style="background-color: yellow" > bold </p >
```

ניתן להגדיר את הגדרות העיצוב עבור כל תגית (element).

הסינטקס של הגדרת העיצוב הינו:

רושמים את התכונה style ולאחריה מוסיפים שווה (=) פותחים גרשיים

כפולים שבתוכם רושמים מאפיין (property) וערך תכונה (value)

שמופרדים בנקודותיים,

ולאחריהם מוסיפים נקודה פסיק ואז ניתן לרשום עוד צימוד של תכונה

וערך שמופרדים בנקודותיים שבסופם נקודה פסיק, וכן הלאה..

בסוף רושמים גרשיים שמעידים על סיום הגדרות העיצוב. סיום כתיבת

הערכים עבור מאפיין style.

דוגמא לסינטקס :

```
< style="property: value; property: value; property:
```

```
value;">
```

סגנון זה נקרא in-line משום שמכניסים אותו בתוך השורה עצמה שבה

רוצים את התוצאה.

סגנון שכזה אינו מתייחס לאלמנטים אחרים בדף זה או באתר בכלל

אלא ספציפית לתגית שבה השתמשנו בתכונת ה- style.

Internal: CSS פנימי

Internal CSS

הינו הגדרת עיצוב כללי עבור דף HTML מסוים.

יצירת תגית מכולה בשם style באזור ה - head עם התכונה

type="text/CSS"

סגנונות פנימיים של מסמך מגדירים סגנונות כלליים לדף HTML הנוכחי ולכל האלמנטים שלו שהוגדרו בתוך התגית style.  
לדוגמא:

הצג דוגמא עם קוד

נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      element {
        color: Green;
        font-family: Verdana;
      }
    </style>
  </head>
  <body>
    <element>hello</element>
    <element>hello</element>
    <element>hello</element>
  </body>
</html>
```

External: CSS חיצוני



## External CSS יצירת קובץ הגדרות עיצוב חיצוני בעל סיומת CSS

וקישורו לדף HTML באזור ה- head מאפשר קביעת סגנונות לאתר שלם.

אילו היית צריך לעבור ולסגן כל פסקה באתר שלך, היית צריך לבצע עבודת נמלים שהייתה נמשכת זמן רב. נהוג ליצור מסמך CSS חיצוני אשר קובע סגנונות כלליים שישמשו את כל האתר.

כל דף HTML המקושר לדף זה מקבל את הסגנונות הכתובים בו. שימוש ב-CSS חיצוני מאפשר למתכנת לשנות את מראה האתר כולו בשינוי של פסקה אחת.

האתר כולו יכול לקבל סגנון אחיד ומיוחד. סגנון חיצוני מקשרים לדף באחת משתי דרכים: קישור או ייבוא.

דוגמא:

יצירת קובץ CSS חיצוני בשם StyleSheet1.CSS קישור הקובץ לדף HTML

```
<head>
<title>Untitled Page</title>
<link type="text/CSS" href="StyleSheet1.CSS"
rel="Stylesheet" />
</head>
```

## ייבוא הקובץ לדף HTML

שימוש בתכונה import הנכתבת בתוך תגית style באזור ה- head. במידה ומשלבים גם עיצוב נוסף בתגית ה style מסוג internal עבור אלמנטים נוספים, יש לכתוב זאת מתחת לשורת הייבוא אחרת הגדרות העיצוב לא יחולו בכל הדפדפנים.

```
<head>
<title>Untitled Page</title>
<style type="text/CSS">
@import URL('StyleSheet1.CSS');
```

&lt;/style&gt;

&lt;/head&gt;



## סדרי עדיפויות

### מה יקרה כאשר יקבע יותר מסגנון אחד לאותו אלמנט?

הערך יקבע על פי העדיפות:

1. הגדרת **CSS inline** - מכריע ללא עוררין.
2. **CSS פנימי (Internal)** - קובע רק אם אין סתירה ב-**inline**.
3. **CSS חיצוני (External)** - קובע רק אם אין סתירה בפנימי או ב-**inline**.
4. **ברירת המחדל**

הגדרת תכונה בשורה עצמה תהיה דומיננטית ותקבע את המאפיין הסופי של הצגת האלמנט גם אם זה מתנגש עם CSS פנימי (הממוקם בתגית ה-HEAD) או חיצוני. CSS פנימי יהיה בעדיפות גבוהה יותר מ-CSS חיצוני וכמובן שבמקרה של אי התאמה, הפנימי יקבע.

## בחירת אופן קביעת הסיגנון

### אז מה כדאי?

לפני שניגש להגדיר את תכונות הסיגנון כדאי שניגע בעוד מספר נקודות. האם ומדוע כדאי להשתמש בסיגנון ולא בתגיות רגילות? CSS מאפשרים הרבה יותר נוחות, גמישות ויעילות בבניית דפי HTML. אולם, בניית דף CSS חיצוני דורש השקעה של זמן ומשתלם לאורך זמן.



אם אתה צריך לבנות אתר של שניים שלושה דפים, לפעמים זה פשוט מיותר. במיוחד אם לא תשתמשו בדפי הסגנון הללו בהמשך.

+

## איזה סוג של CSS, חיצוני פנימי או inline?

-

### חיצוני - בעד ונגד

גיליונות סגנון עוזרים לך לעצב באופן אחיד אתרים שלמים ולכן הם המומלצים ביותר. אם יש שינוי סגנון, כתוב אותו בדף ה-CSS ומיד הוא מיושם לרוחב האתר כולו. הבעיה עם קבצי CSS חיצוניים היא הזמן שלוקח לדפדפן לפתוח אותם בזמן טעינת דף. יש צורך לטעון לא רק את הדף עצמו אלא גם את גיליון הסגנון עוד בטרם יוצג הדף למשתמש. לכן יש צורך לשמור את קבצי ה-CSS קטנים.

### פנימי - בעד ונגד

סגנון פנימי יהיה שימושי במיוחד כאשר רוצים ליצור מסמך מיוחד השונה במראהו משאר האתר. ניתן ליצור כללי עיצוב ספציפיים לדף מסוים. הצרה עם סגנונות פנימיים היא שיש נטייה ליצור כללים לכל דף בנפרד ולאחר מכן, להעתיק את כללי העיצוב לדף הבא באתר ואילו לדפים נוספים. הבעיה נוצרת כאשר יש צורך לשנות משהו (ותמיד תדרשו לבצע שינויים באתר) ואז המלאכה הופכת אימתנית וטעונת שגיאות. כלל הברזל הוא, כשיש כלל סגנון הנוגע לשלושה דפי HTML או יותר, יש להעביר אותו לקובץ CSS ולקשר או לייבא אותו למסמך. כלל זה יעזור מאוד כשיגיע הזמן לשינויים.

### Inline - בעד ונגד

סגנונות inline הם בזבזניים, קשים לניהול ואינם ניתנים לשימוש חוזר ולכן מומלץ להשתמש בהם כמה שפחות. הם קשים מאוד לתחזוקה שוטפת ושינויים תכופים של עיצוב האתר.

דפי HTML שבהם כתובים סגנונות רבים ב-inline, הופכים להיות בדיוק מה שלא רצינו שיהיו: עיצוב, תוכן ותכנות בעירבוביה.

ולכן בכל פעם שאתה מתפתה לשים סגנון בתוך ה body, תחשוב שוב  
אם אינך יכול לעשות זאת באופן שונה,  
בין אם שימוש ב CLASS או ב ID. הסגנונות יהיו הרבה יותר נגישים  
לסגנון ועיצוב.

+

-

## יבוא קובץ CSS חיצוני

קובץ CSS כותבים כקובץ טקסט רגיל (כמו דפי HTML) ב-Notepad  
או בכל עורך אחר. כל הטקסט בנוי במתכונת הבאה:

```
p
{
  color:Blue;font-size:larger;text-align:center;
  text-decoration:underline;
}
a
{
  font-family:Verdana;font-weight:bold;
}
li
{
  list-style-type:none;padding:0px;color:Navy;
}
```

את קובץ הטקסט הזה שומרים עם סיומת CSS.  
כדי שדף ה HTML שלך ידע לקבל הגדרות CSS שהגדרת בקובץ  
חיצוני צריך לייבא קובץ CSS לתוך דף ה HTML שלך.  
זאת עושים באחת משתי דרכים:

### 1. יבוא באמצעות הפקודה @import בתוך

### תגית <style>

יבוא מיושם גם הוא בתוך חלק ב- <head> אבל תמיד בתוך תגית

:<style>

+

```
<style type="text/CSS">
```

```
@import URL('StyleSheet1.CSS');
```

```
@import URL('StyleSheet2.CSS');
```

```
h1{text-align:center;}
```

```
</style>
```

```
</head>
```

-

במקרה זה יובאו שני דפי סגנון

הפקודה @import חייבת להיכלל לפני כל הגדרת סגנון פנימית

אחרת. אם תנסה להכניס את הפקודה אחרי הגדרות סגנון

אחרות, יתעלם הHTML הקובץ הCSS המוכלל.

את פקודת ה - @import ניתן להכליל גם בתוך קבצי CSS

חיצוניים וכך לקונן דפי סגנון.

ניתן ליצור קובץ CSS חיצוני ובתוכו לייבא מספר קבצי CSS

נוספים ואת הקובץ הראשי לייבא או ליצור עבור <link> לכל דף

באתר,

כך ניתן לשלוט על עיצוב האתר בדרך יעילה של חלוקת האתר

לחלקים

ואף ניתן להגדיר באחד מהקבצים שמייבאים עיצובים שלא

נתמכים בדפדפנים ישנים וכך להימנע מלהציגם.

דפדפנים ישנים לא מכירים את פקודת Import ולכן הם יתעלמו

משורה זו.

אפשר ליחד קבצי CSS למדיה מסוימת בזמן היבוא באופן הבא

וכך, רק עם הדפסתם או הצגת הדף בברייל ייושמו הגדרות דף

הCSS המיובא:

```
@import "StyleSheet1.CSS" print,braille;
```

## 2. קישור באמצעות תגית <link>



קישור, מיושם תמיד בתוך מתחם שבין פתיחת תגית <head> לסגירתה, כך:



```
<head>
  <title>Untitled Page</title>
  <link type="text/CSS" href="StyleSheet2.CSS"
rel="Stylesheet" />
  <link type="text/CSS" href="StyleSheet1.CSS"
rel="Stylesheet" />
</head>
```

במקרה זה יובאו שני דפי סגנון

שימו לב שדף הסגנון השני ידרוס בהגדרותיו את ההגדרות של הדף הראשון אם תהיה התנגשות בהגדרות, מעצם היותו מאוחר יותר בסדר ביצוע הפעולות.

לעומת זאת, התגית <h1>, אפילו אם הוגדרה בשני דפי ה-CSS החיצוניים כבעלת צבע ירוק תיצבע בצבע אדום כי לפי סדר העדיפויות, הגדרה בדף עצמו היא הקובעת.

## מרכיבי השפה

עיצוב עבור מספר תגיות

גיליונות סגנון בין אם מדובר בחיצוניים או פנימיים, (לא inline) יש תחביר מאוד פשוט:

Selector - אלמנט ה - HTML אליו מתייחסים או שם כללי לסגנון.

Property - התכונה אותה רוצים להגדיר.

## Value - הערך שצריך לתת לתכונה.

לדוגמא: +

Selector { Property:Value; } -

לדוגמא, כאשר רוצים לקבוע צבע רקע כחול לכל הטבלאות יש לכתוב זאת כך:

```
table{background-color:Blue;}
```

ניתן לכתוב מספר תכונות שונות לאותו סלקטור.  
בין התכונות יש להפריד באמצעות semicolon (הידוע בכינויו נקודה-פסיק).

```
table{background-color:Blue;width:50px;}
```

כדי שהמסמך יהיה קריא יותר, אפשר לכתוב כל תכונה בשורה נפרדת:

```
table
{
background-color: Blue;
width: 50px;
}
```

## עיצוב עבור מספר תגיות

ניתן גם לתת תכונות משותפות לכמה סלקטורים שונים.  
בדוגמא הבאה ניתן צבע גופן אדום לכותרות גדולות <h1> וכן  
לפסקאות <p>  
את הסלקטורים יש להפריד בפסיקים:

```
h1,p
{
color:Red;font-family:Verdana;
}
```

CLASS

+

התכונה class מאפשר לנו ליצור שמות לסוגי סגנונות עיצוב ולהחיל אותם על תגיות מסוימות לפי בחירתנו.

-

שם של class תמיד יתחיל בסימן נקודה ולאחריו שם באנגלית שאנו נמציא,

כל שם שנגדיר יוכל לעבוד כל עוד השם אינו זהה לשם של תגית מסוימת או למילה שמורה במערכת שמשתמשים בה לצרכים אחרים. לדוגמא:

כאשר ניצור בתוך תגית ה-style הגדרת עיצוב עבור festi.

כרגע הסגנון הזה אינו פעיל במסמך.

אולם בכל תגית שבה תוכנס השורה:

class="festi"

הטקסט שבתוכה יצבע באפור,  
ייקבל שוליים שמאליים באורך 10 פיקסלים,  
וימתח תחתיו קו תחתני נטוי.

<    הצג דוגמא עם קוד

<    נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      .festi{
        color:Gray;
        margin-left:10px;
        text-decoration:underline;
        font-style:italic;
      }
    </style>
  </head>
  <body>
    <p class="festi">this is my paragraph</p>
```

</body>

</html>



יצירת class עבור סוג מסוים של תגית

ניתן ליצור class שנוכל להחילו רק עבור תגית מסוג מסוים.  
לדוגמא כאשר נרצה ליצור class רק עבור תגיות מסוג td נכתוב זאת כך:

```
td.Header
{
font-weight:bold;
text-align:center;
}
```

יש לרשום את שם התגית הרלוונטית אחריה נקודה ואז שם ה class שקבענו  
נוכל להשתמש ב class זה רק בתגית td, עבור שאר התגיות לא תהיה שום השפעה.  
בדוגמא הבאה קיימות הגדרות סותרות עבור אותו אלמנט

<    הצג דוגמא עם קוד

<    נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      p {
        color: Red;
      }

      p {
```

```
        color: Blue;
    }
</style>
</head>
<body>
    <h1>headline</h1>
    <p>first</p>
    <p>second</p>
</body>
</html>
```



כאשר קיימת סתירה בין פקודות CSS בעלות עדיפות זהה, הפקודה האחרונה היא הקובעת ולכן שתי הפסקאות יצבעו בכחול בלבד. לדוגמא: הגדרת סגנון עיצוב לתגית P וגם עבור class מסוים .

```
.myClass ,p
{
color:Blue;
text-decoration:underline;
}
```

העיצוב יחול במסמך על כל הפסקאות ועל כל התגיות אשר משתמשות ב class. לדוגמא: הגדרת עיצוב עבור class , עבור תגית מסוימת ועבור class שניתן להחילו רק על תגית ספציפית.

```
.myClass ,p , span.myclass2
{
color:Blue;
text-decoration:underline;
}
```

העיצוב יחול במסמך על כל הפסקאות ועל כל התגיות אשר משתמשות ב class. בנוסף יחול העיצוב על כל תגיות ה span אשר משתמשות במחלקה myclass2.



הגדרת סגנון לפי מאפיין ID

ניתן להגדיר מאפיין ID עבור כל תגית.  
ID הינו מאפיין ייחודי עבור כל תגית ואי אפשר להשתמש באותו ערך ID יותר מפעם אחת בכל דף HTML.  
לכן מתן סגנון עיצוב לפי מאפיין ID הינו כמעט כמו לתת סגנון inline עבור תגית מסוימת, אלא שבדרך כלל נוח יותר לעשות זאת דרך תגית ה – style.

כדי להגדיר סגנון ל ID יש לכתוב סולמית ולאחריה את ה- ID של האלמנט (בצמוד).

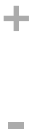
לדוגמא:

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה <

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #first {
        text-align: center;
        text-decoration: overline;
      }
    </style>
  </head>
  <body>
    <h1 id="first">headline</h1>
    <p>first</p>
    <p>second</p>
    <h1>headline2</h1>
```

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



בדוגמא הוגדר מאפיין ID עבור תגית h1:

```
id="first"
```

והוגדר עיצוב ייחודי שמשפיע רק עבור תגית זאת ספציפית.  
הטקסט יהיה ממורכז עם קו עליו.

שילוב של מספר הגדרות

לדוגמא : הגדרת עיצוב עבור מספר תגיות בעלות id שונה וגם יצירת class

```
#list2, #list3 , .myClass
{
color:Red;
}
```

יש לרשום פסיק בין הגדרת עיצוב לתגית מסוימת לבאה אחריה.

כתיבת הערות

ב-CSS ניתן לכתוב הערות כדי לעזור לקורא הקוד להבין מדוע ואיך נכתבה התוכנית.  
בכתיבת הערות יש לתחום אותם משני הצדדים ב  
/\* הערות כאן \*/  
כדי שהתוכנית לא תנסה להתייחס למידע הכתוב בהערות.  
אפשר שורה אחת או מספר שורות. כך:

```
<style type="text/CSS">
    /* הגדרת עיצוב עבור תגית מסוימת */
    #first {
        text-align: center;
        text-decoration: overline;
    }
```

.festi {  
 הגדרת מחלקת עיצוב /\*  
 \*/אשר נשתמש בה לעיצוב החלק הימני של האתר  
}  
</style>

חשוב מאוד לסגור הערות מפני שאם לא תסגרו אותן, הדפדפן יתייחס לשאר התוכנית כאל הערה ולא יבצע את הקוד.

inheritance – הורשה

הורשה הינו מינוח פשוט למדי.  
לדוגמא אנחנו מעוניינים להגדיר עבור כל האלמנטים בדף צבע כתב כחול במקום לרשום עיצוב עבור כל אחת מהתגיות בשורה שנראית כך:

p,h1,h2,il,td  
{  
color:Blue;  
}

ניתן להגדיר לתגית body צבע כתב והתגית body תוריש את סגנון העיצוב לכל התגיות שבדף.  
הגדרת עיצוב לאלמנט ראשי תחול על כל האלמנטים שמקוננים בתוכו.

body  
{  
color:Blue;  
}

siblings

Child selector – תגית "ילד" (בן)	+
Descent selector – תגית "אב" (הורה)	
הגדרת סגנון עיצוב עבור תגית "ילד" (child) הנמצאת בתוך תגית חיצונית העוטפת אותה (descent) , הגדרת עיצוב אחיד לכל התגיות המקוננות.	-

לדוגמא הקוד הבא:

< הצג דוגמא עם קוד

< נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #ul2 > li
      {
        color:Red;
      }
    </style>
  </head>
  <body>
    <ul>
      <li>1</li>
    <ul>
      <li>11</li>
      <li>12</li>
      <li>13</li>
    </ul>
    <li>2</li>
    <ul id="ul2">
      <li>2</li>
      <li>
        3
      <ul>
```

<li>31</li>

<li>32</li>

<li>33</li>

</ul>

</li>

<li>4</li>

</ul>

</ul>

</body>

</html>

הוגדר מאפיין מזהה id עבור אחת מתגיות ה <ul> ועיצבנו את התגיות המקוננות שבתוכה בצבע אדום. על ידי

1. כתיבת תגית ה"אב" (במקרה זה רשמנו את ערך ה id של תגית האב),

2. סימן גדול מתגית ה"ילד"

3. כתיבת תגית ה"ילד"

הסימן השוואה אומר לדפדפן שיתייחס בפענוח הפקודה רק לתגית li הנמצאת בתוך תגית עם מאפיין id="ul2". ניתן היה להגדיר עיצוב רק עבור תגית האב (id="ul2") ובכל מקרה העיצוב היה חל על כל התגיות <li> אשר מקוננות בתוכה, מכיוון שתמיד קיימת הורשה בין התגית הראשית לתגיות המקוננות. כמו שהגדרנו עיצוב לתגית body שחל על כל האלמנטים בדף.

הגדרות סגנון קונטקסטואליות

הגדרת סגנון עיצוב עבור תגיות לפי מיקומן בתוך תגיות אחרות. ניתן להגדיר עיצוב עבור אלמנט מסוים אבל שאותו אלמנט יושפע מהעיצוב רק אם הוא ממוקם בתוך אלמנט אחר. (הכוונה רק אם אלמנט מסוים מקונן בתוך אלמנט אחר).

לדוגמא:  
קיימות תגיות <a> בתוך תאים בטבלה וכמו כן קיימות תגיות <a> בעוד מקומות בדף,

https://t2k-pap-prod-assets.s3.eu-west-1.amazonaws.com/public/e6397bca-cda9-4986-86d9-992a34885a9c/b36c071d-e6a2-4901-8393-b390467170c1/t2k-u...

21/105

המטרה לעצב את כל תגיות הקישור שבתוך הטבלה בעיצוב מסוים וכל  
שאר תגיות הקישור שממוקמות באזורים אחרים בדף בעיצוב שונה.

+

הצג דוגמא עם קוד

<

-

נסה אותי בתצוגה מלאה

<

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      a
      {
        color:Green;
      }
      table a
      {
        color:Fuchsia;
      }
    </style>
  </head>
  <body>
    <a>a1</a>
    <table border="1">
      <tr>
        <td>
          <a>a2</a>
        </td>
      </tr>
    </table>
  </body>
</html>
```

בקוד שלפניכם כל הקישורים בדף יצבעו בצבע ירוק למעט הקישורים שבתוך תגית <table> אשר יצבעו בצבע ורוד.  
יש לרשום את התגית הראשית ולאחריה רווח ואז התגית שרוצים להגיד עבודה את העיצוב הקונטקסטואלי.

דוגמה נוספת: +

הגדרת עיצוב עבור כותרות המשמשות כקישור, הכוונה תגית <h1>

המקוננת בתוך תגית <a> -

< הצג דוגמא עם קוד

< נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      a h1
      {
        color:red;
      }
    </style>
  </head>
  <body>
    <h1>click</h1>
    <a><h1>here</h1></a>
  </body>
</html>
```

דוגמה נוספת:

הגדרת עיצוב עבור רשימות, קיימות כאן שלוש רמות של רשימות

פריטים, ראשיים , משניים ותתי משניים.

```
ol li
{
color:Blue;
list-style-type:lower-greek;
}
ol ol li
```

```
{
color:Red;
list-style-type:upper-alpha;
}
ol ol ol li
{
color:Green; }
```

+

-

עבור כל אחת מהרמות הוגדרו הגדרות עיצוב שונות בהתאם למיקום:

- ol ol אלו הפריטים הראשיים
- li ol ol אלו תתי הפריטים שמפרטים את הפריטים הראשיים
- li ol ol ol אלו תתי הפריטים שמפרטים את הפריטים המשניים.

## הגדרת מיקום לתגית מסוימת לפי מיקומה ללא קשר לקיבון תגיות

לדוגמא:

<    הצג דוגמא עם קוד

<    נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      h1 + p
      {
        color:Fuchsia;
      }
    </style>
  </head>
  <body>
    <h1> line 1 </h1>
    <p> line 2 </p>
```



```
<p> line 3 </p>
<h1> line 4 </h1>
<p> line 5 </p>
<p> line 6 </p>
</body>
</html>
```

הגדרה זו מעצבת את כל תגיות <p> הנמצאות ישר לאחר תגיות <h1> בכל המסמך  
מדובר רק על התגית <p> הראשונה שנמצאת לאחר תגית <h1> ללא שום תגית הכתובה ביניהם.  
עיצוב זה יחול בכל המקומות במסמך ששם מסודרות התגיות בסדר שנקבע בהגדרת הסגנון.

עיצוב לכל הדף

הגדרת סגנון של הסימן \* תתייחס לכל האלמנטים בדף ה HTML.  
כמובן שהעיצוב היותר ספציפי במקרה של סתירה הוא זה שיקבע.

< הצג דוגמא עם קוד

< נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      *
      {
        color:Yellow;
      }
    </style>
  </head>
  <body>
    <h1> line 1 </h1>
```

```
<p> line 2 </p>
<p> line 3 </p>
<h1> line 4 </h1>
<p> line 5 </p>
<p> line 6 </p>
</body>
</html>
```

סימון הכוכבית קרוי לעיתים גם "joker" היות שהוא חל על כל התגיות במסמך ועוזר במקרה בו פספסנו להגדיר עיצוב במקומות מסוימים בדף.

תתי - אלמנטים

קיימים בדף HTML מספר אלמנטים שאי-אפשר להגדירם בתוך תגית מסוימת, כגון: אות ראשונה בפסקה או שורה ראשונה בפסקה.

p:first-letter

הדוגמה הבאה מגדירה לדפדפן להציג את האות הראשונה בפסקה בגודל כפול, ולהטותה הצידה:

הצג דוגמא עם קוד

נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      p:first-letter
      {
        font-size:30px;font-style:italic;
```

+

-

## p:first-line

<p>

>

 $\wedge$ 

<https://t2k-pap-prod-assets.s3.eu-west-1.amazonaws.com/public/e6397bca-cda9-4986-86d9-992a34885a9c/b36c071d-e6a2-4901-8393-b390467170c1/t2k-u...> 27/105

line x<br/>

line y<br/>

</p>

</body>

</html>

+

-

קיימים גת תתי אלמנטים עבור תגית <a>

a:link

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה <

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      a:link
      {
        color:Fuchsia;
      }
    </style>
  </head>
  <body>
    <a href="https://www.johnbryce.co.il/">go to
  </body>
</html>
```

עיצוב הקישור טרם ביקרנו בו, לדוגמא:

a:visited

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה

< +

-

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      a:visited
      {
        text-decoration:none;
      }
    </style>
  </head>
  <body>
    <a href="https://www.johnbryce.co.il/">go to
  </body>
</html>
```

עיצוב הקישור שביקרנו בו, לדוגמא:

a: hover

עיצוב קישור בעת ריחוף העכבר מעל האלמנט, לדוגמא:

הצג דוגמא עם קוד

<

נסה אותי בתצוגה מלאה

<

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      a: hover
      {
        font-size:30px;
      }
    </style>
```

```
</head>

<body>

    <a href="https://www.johnbryce.co.il/">go to

</body>

</html>
```

## a:active

– עיצוב הקישור כאשר לוחצים עליו והקישור הופך להיות פעיל (בזמן הלחיצה)  
לדוגמא:

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה <

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

    <head>

        <style type="text/CSS">

            a:active

            {

                color:Green;

            }

        </style>

    </head>

    <body>

        <a href="https://www.johnbryce.co.il/">go to

    </body>

</html>
```

## focus

קיים גם תת אלמנט הקרוי focus ,  
אשר מגדיר עיצוב כאשר הפוקוס נמצא באלמנט מסוים, בלחיצת עכבר או במעבר בעזרת מקש TAB.

תת אלמנט זה עדיין אינו פועל עבור כל התגיות.

לדוגמא:

+

הצג דוגמא עם קוד

<

-

נסה אותי בתצוגה מלאה

<

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      input:focus
      {
        background-color:Yellow;
      }
    </style>
  </head>
  <body>
    <input type="text" />
    <br />
    <input type="password" />
    <br />
    <input type="password" />
  </body>
</html>
```

כאשר נלחץ בתוך תגית input צבע הרקע של הפקד ישתנה לצהוב ,  
ביציאה מהפקד צבע הרקע יחזור להגדרתו הקודמת.

hover

תת אלמנט hover פעיל עבור כל התגיות ומשנה את עיצובן בעת ריחוף  
העכבר מעל כל אלמנט שהגדרנו במסמך  
לדוגמא שינוי רוחב תמונה:

הצג דוגמא עם קוד

<

נסה אותי בתצוגה מלאה



```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      img:hover
      {
        width:150px;
      }
    </style>
  </head>
  <body>
    <img src='data:image/jpeg;base64,/9j/4AAQSkZJRgA
  </body>
</html>
```

ניתן גם ליצור תת אלמנט עבור מחלקה.  
</img> יצרנו מחלקה עבור תגית

הצג דוגמא עם קוד



נסה אותי בתצוגה מלאה



```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      .myimg1
      {
        width:100px;
        height:100px;
      }
    </style>
  </head>
```



```
<body>

  <img class="myimg1" src='data:image/jpg;base64, +
</body>

</html>
```

ניתן ליצור עיצוב שיחול על התגית רק במעבר עכבר, אבל עיצוב ספציפי לתגית שחל עליה סגנון ה class שיצרנו רושמים את שם המחלקה נקודותיים hover לדוגמא:

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה <

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <head>
      <style type="text/CSS">
        .myimg1:hover
        {
          width:500px;
        }

      </style>
    </head>
    <body>
      <img class="myimg1" src='data:image/jpg;base64,
    </body>

  </html>
```

## עיצוב עבור תגיות הנמצאות בתוך אלמנט מסוים

לדוגמא <div> עם מאפיין "id="right" כתיבת עיצוב עבור Id רווח שם התגית:

הצג דוגמא עם קוד

<+  
-

נסה אותי בתצוגה מלאה

<

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #right p
      {
        color:Fuchsia;
      }
    </style>
  </head>
  <body>
    <div id="right">
      <p>first</p>
    </div>
    <p>second</p>
  </body>
</html>
```

ניתן גם ליצור עיצוב שיחול על התגית רק במעבר עכבר מוסיפים ערך של תת אלמנט אחרי שם התגית הרצויה לדוגמא:

הצג דוגמא עם קוד

<

נסה אותי בתצוגה מלאה

<

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
```

```
#right p:hover
{
    background-color:gray;
    width:30px;
}
</style>
</head>
<body>
    <div id="right">
        <p>first</p>
    </div>
    <p>second</p>
</body>
</html>
```

עיצוב תגיות form

ניתן להחיל עיצוב אחיד לכל תגיות הטופס בעזרת הגדרת עיצוב עבור תגית input לדוגמא:

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה <

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      input{
        width:200px;
      }
    </style>
  </head>
```

```
<body>
  <input type="text" />
  <br />
  <input type="password" />
  <br />
  <input type="password" />
</body>
</html>
```

ניתן גם להגדיר עיצוב ספציפי יותר לפי תכונת ה type  
דוגמא עבור תיבת טקסט:

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה <

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      input[type="text"]
      {
        background-color:Yellow;
      }
    </style>
  </head>
  <body>
    <input type="text" />
    <br />
    <input type="password" />
    <br />
    <input type="password" />
  </body>
</html>
```

דוגמא עבור תיבת סיסמא:

הצג דוגמא עם קוד

<+<-

נסה אותי בתצוגה מלאה

<

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      input[type="password"]
      {
        background-color:Blue;
      }
    </style>
  </head>
  <body>
    <input type="text" />
    <br />
    <input type="password" />
    <br />
    <input type="password" />
  </body>
</html>
```

דוגמא לשילוב הגדרות עיצוב עבור מספר תתי אלמנטים:

הצג דוגמא עם קוד

<

נסה אותי בתצוגה מלאה

<

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      input[type="text"] , input[type="password"
```

```
{
  background-color:Yellow;
}
</style>
</head>
<body>
  <input type="text" />
  <br />
  <input type="password" />
  <br />
  <input type="password" />
</body>
</html>
```

כמו תמיד יש להוסיף פסיק שיפריד בין שיוך עיצוב אחד לשני.

הערה: אסור להוסיף רווח בין שם התגית לתכונת ה type או בין שם התגית לתת אלמנט, רווחים במקרים של שימוש בתתי אלמנטים משבשים את הקוד.

ניתן גם להחיל עיצוב של תת אלמנט focus.  
לדוגמא:

<    הצג דוגמא עם קוד

<    נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      input:focus,textarea:focus
      {
        background-color:Gray;
      }
    </style>
  </head>
  <body>
```

```
<textarea></textarea>
<br />
<input type="text" />
<br />
<input type="password" />
<br />
<input type="password" />
</body>
</html>
```

כאשר הפוקוס יהיה בתוך אחד מהפקדים צבע הרקע ישתנה לאפור, במעבר לפקד אחר במסך צבע הרקע יחזור להגדרת הצבע שהוגדרה עבור אותו פקד.

סוגי פקדים נוספים:

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה <

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      fieldset
      {
        width:200px;
        background-color:Gray;
      }
    </style>
  </head>
  <body>
    <fieldset>
      <input type="text" />
      <br />
      <input type="password" />
```

<br />

<input type="password" />

+</fieldset>

</body>

-</html>

תרגילים

תרגיל 1

צרו דף עם גלריית תמונות , בכל שורה באתר יש להוסיף 3-4 תמונות.  
לכל שורת תמונות יש להגדיר גודל תמונה זהה בעזרת class מתאים  
(גובה ורוחב).  
יש ליצור עיצוב שמופעל במעבר עכבר מעל לתמונה, יש ליצור עיצוב  
שונה עבור כל מחלקה.

יחידות מידה עבור סגנונות

סוגי יחידות המידה

סוגים של ערכים שניתן לתת למאפייני הסגנון:

1. אורכים
2. כתובות (URLים)
3. צבעים
4. מילות מפתח המגדירות את עיצוב הסגנון למשל **bold**



## אורכים

אורכים מתייחסים לגדלים – גודל גופן/תמונה. טבלה, רוחב, גובה, שוליים, מרחק אלמנט מסוים מגבול של אלמנט אחר, עובי גבול וכו'. לאורכים ניתן להוסיף גם את סימן ה + או - כדי לסמן הוספה או הפחתה מהערך המיידני של התכונה, וכן יש לצרף, ללא רווחים, שתי אותיות המסמנות את יחידת המידה או את סימן האחוזים. כמו כן ניתן לקבוע אורך קבוע מבלי להתייחס לאורך שהיה קיים.

### px, cm, mm, pt, pc, em, ex, in

- in – אינצ'ים, 1 אינץ' שווה 2.54 ס"מ
- ex – גודל האות X.
- em – גודל יחסי, היחס הינו לעומת גודל הכתב של האלמנט העוטף.
- pc – פיקה, 1 פיקה שווה 12 נקודות.
- pt – נקודות (1/72 ס"מ).
- mm – מילימטרים.
- cm – סנטימטרים.
- px – פיקסלים.

### יש הבדל אחד עיקרי בין ארבע Px, pt, em, %

היחידות 'אפ' ו-'פט' קובעות לגופן גודל מוחלט, בעוד ש-'%' ו-'em' מאפשרות למשתמש לשנות את גודל הגופן כפי שיבחר. ישנם משתמשים מוגבלים, מבוגרים או כאלו שפשוט סובלים מראיה לקויה או מצג מחשב באיכות גרועה וכדי שהאתר שלך יהיה נגיש לכולם ניתן להשתמש ביחידות שניתנות לשינוי כ-'%' או 'em'.

כתובות

ערכי URL חייבים לרשום בתוך סוגריים רגילים מייד אחרי המילה URL.  
את ערך ה URL יש לעטוף מסביב עם גרשיים כפולים או גרש בודד מסביב.  
לדוגמא:  
צבע רקע מסוג תמונה לתגית body

```
body{background-image:URL('ballon1.jpg');}
```

תכונות של צבעים

ערכים של צבעים יכולים להיכתב במספר דרכים:

- 1. שם הצבע
- 2. ערכי RGB
- 3. ערך הקסדצימלי
- 4. ערך עשרוני

לדוגמא:

```
color:Red;  
color:rgb(0,30,250);  
color:#00ff33;  
color:rgb(0%,100%,0%);
```

שלישיית RGB מגדירה לכל אחד מהצבעים – red ,green, blue גוון צבע עם ערכים שנעים בין 0-255.

- הערך 0,0,0 מייצג את הצבע השחור
- הערך 255,255,255 הינו לבן

מילות מפתח



הגדרת ערך של סגנון לא בעזרת ערך מספרי אלא בעזרת מילה שמורה  
המייצגת סוג סגנון  
לדוגמא:

```
p
{
  font-weight:bold;
}
```

הגדרת סגנון טקסט מודגש בעזרת הערך השמור `bold`.

תרגילים

סגנונות עיצוב

עיצוב גופנים

Font-family

סוג הגופן: רשימה של סוגי גופנים מופרדים בפסיקים.  
נהוג לרשום מספר גופנים מפני שלא בכל מחשב מותקנים כל הגופנים.  
הדפדפן משתמש בגופן הראשון מתוך הרשימה שמותקן גם במחשב הלקוח.  
אם אף אחד מהגופנים המבוקשים לא קיים, ישתמש הדפדפן בגופן ברירת המחדל של הדפדפן.

מסיבה זו, כדאי לבדוק את תצוגת הדף עם מספר פונטים רבים ככל

האפשר ולמצוא ולכתוב לפי הסדר את המתאימים ביותר.  
שיוצרים רשימת גופנים לאתר האינטרנט באופן טבעי נתחיל עם  
הגופנים המתאימים ביותר לעיצוב שבחרנו ונמשיך עם חלופות קרובות  
פחות ופחות.



מומלץ מאוד לסיים את הרשימה עם משפחה גנרית.  
בצורה זו לפחות נבטיח שהדף יוצג עם גופן מאותה משפחה אם הגופן  
המסוים אינו זמין.

ישנם 3 סוגי משפחות גנריות:

1. Serif – מאופיינות בכך שלמשפחה יש "רגליים".

גופנים מהרשימה :

;font-family:Times New Roman,Garamond,Georgia

2. sans serif - מאופיינות בכך שלמשפחה יש " אין רגליים".

גופנים מהרשימה:

;font-family:Trebuchet,Arial,Verdana

3. monospace – לכולם יש רוחב תווים קבוע.

גופנים מהרשימה:

;font-family:Courier,Courier New

דוגמא לשימוש במספר גופנים:

font-family:Times New Roman,Arial,Verdana,Sans-Serif;

חשוב לרשום את הסימן פסיק כמפריד בין סגנון עיצוב אחד לשני

## font-size

גודל גופן , ניתן להגדיר את הגודל כאבסולוטי או יחסי.  
יחידות המידה יכולות להיות : יחידות מידה יכולות להיות: px, cm,  
mm ,pt ,pc ,em ,ex ,in  
לדוגמא:

font-size:20xp;

font-size:20pt;

font-size:2em;

font-size:120%;

יש גם אפשרות לגודל קבוע שנקבע ע"י הדפדפן וניתן להשתמש בו  
באחד מהערכים הבאים: +  
xx-large, xx-small, x-small, small, medium, large, x-large  
אלו מותאמים לגדלי הכותרות הנפוצים. <7-1H> -

## font-style

סגנון הכתב יכול לקבל אחד משלושה ערכים: italic, oblique, normal.  
Normal הינו כתב רגיל ואילו בין שני האחרים קשה להבדיל, שניהם נטויים.

## Font-weight

עיצוב טקסט כמודגש, תכונה המאפשרת לעבות את האותיות.  
ערך ברירת המחדל הוא normal  
אולם ניתן לתת את הערך bold כדי לקבוע אות עבה או להשתמש  
בערכים היחסיים bolder או lighter באופן יחסי לאלמנט ההורה.  
במקום מילים, אפשר להשתמש במספרים: 100 הוא העדין ביותר, 900  
הוא המודגש ביותר. 400 הוא הערך הרגיל, ו-700 הוא הערך הקבוע  
כ-bold.

font-weight:bold;

## Line-height

היא גובה שורת האותיות, ומיוצגת באמצעות מידת אורך כמוזכר לעיל.  
לתכונה זו אפשר לתת ערכים מספריים, אחוזים, או יחסיים.  
תכונה זו מאפשרת ליצור רווח בין שורות.

line-height:+10px;

line-height:-10px;

line-height:20px;

## font-variant

רלוונטי רק עבור טקסט לטיני. יכולה לקבל שני ערכים: normal ו-  
small-caps.  
האפשרות האחרונה הופכת את האותיות הכתובות לאותיות גדולות (מ)  
(לA).

font-variant:small-caps;



## font-stretch

תכונה אחרת ופחות מוכרת (משום שרוב הדפדפנים אינם תומכים בה עדיין) היא font-stretch. תכונה זו מאפשרת להציג את האותיות מכווצות או רחבות לפי בחירתך. ניתן לשים את הערכים:

Ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded. Font-size-adjust.

היא תכונה המעניקה פרופורציה שונה של יחסי רוחב/גובה לאותיות. תכונה זו עדיין לא נתמכת ברוב הדפדפנים. פעמים רבות אנו נותנים ערכים ליותר מתכונה אחת של גופן, יש לרשום את כל הערכים בסדר מסוים עם רווח ביניהם:

font-style | font-weight | font-size | font-family

לדוגמא:

font:italic bolder 30px verdana;

יש להוסיף רווח המפריד בין התכונות ולא פסיקים.

## color

צבע כתב , הפקודה מקבלת ערכי צבעים לפי שם צבע, שילוב גוונים של RGB, ערך צבע הקסדצימלי ושילוב גוני RGB באחוזים.

לדוגמא:

color:Red;  
color:rgb(0,30,250);  
color:#00ff33;  
color:rgb(0%,100%,0%);

## עיצוב טקסט



תכונות טקסט מגדירות כיצד טקסט ממוקם בדף וכיצד הוא מוצג למשתמש.

### Letter-spacing

מוסיפה רווחים בין האותיות.

```
letter-spacing:5px;
```

תכונה שימושית במיוחד כאשר רוצים להדגיש חלקי טקסטים חשובים.

### Text-align

מגדיר לאיזה כיוון הטקסט יתיישר בהתאם לדף ימין, שמאל, מרכז או צדדים. right, left, center, או justify, בהתאמה. מומלץ להשתמש בתכונה זו של CSS ולא בתכונות ה-align של תגיות רגילות.

```
text-align:center;
```

תכונה זו פועלת רק על תגיות מסוג block-element.

### Text-decoration

מקבל את אחת או יותר מהתכונות הבאות: underline, line-through, blink, none, overline. באמצעות תכונה זו ניתן לאמר לדפדפן להדגיש בצורה מסוימת את הטקסט. אפשר לתת יותר מערך אחד:

#### הוספת קו תחתי

```
text-decoration:underline;
```

#### הוספת קו עילי ותחתי

```
text-decoration:overline underline;
```

blink עדיין לא נתמך בכל הדפדפנים.

## text-indent

הגדרת כניסה בפסקה עבור השורה הראשונה , מבליט את השורה הראשונה.

text-indent:20px;

## text-transform

עיצוב האותיות באנגלית  
שינוי מאותיות גדולות לקטנות (uppercase) וההיפך (lowercase) או להגדיל רק את האות הראשונה בכל מילה (capitalize).  
אין שום השפעה על אותיות בעברית.

text-transform:capitalize;

## vertical-align

שולטת במיקום האנכי של אובייקטים. הערכים הקבילים לתכונה זו הם:

baseline, middle, sub, super, text-top, bottom, top.

## word-spacing

בדומה לletter-spacing מוסיפה רווחים בין מילים.  
ערך ברירת המחדל כאן הוא normal, ואפשר לכתוב ערכי גודל במספרים, אחוזים או ערכים יחסיים

word-spacing:10px;

## direction

הגדרת כיוון הטקסט, מימין לשמאל או משמאל לימין

direction:rtl;

עיצוב גבולות



כמעט לכל אלמנט ניתן ליצור גבול , לגבולות יש מספר תכונות שניתן לשנות כגון צבע סגנון ועובי.

+

ניתן לקבוע תכונות עבור הפאות השונות של הגבול – מעלה, מטה ימין ושמאל.

-

אם רושמים ערך תכונה אחת כל הפאות של הגבול יעוצבו אותו דבר , שני ערכים על הפאות העליונות והתחתונות יחול העיצוב הראשון ועל הפאות משמאל ומימין יחול העיצוב השני. שלושה ערכים – הראשון לפאה העליונה , השני לפאה הימנית והשמאלית והשלישי לפאה התחתונה.

## border-style

סגנון הגבול, הערכים הינן מילים שמורות המציינות איזה גבול יוצג - רציף קוי , נקודות ועוד.

border-style:solid;

יצירת גבול עם מספר סגנונות לפי הפאות השונות כיוון הפאות הוא מתחיל מהפאה העליונה בסדר השעון – עליון, ימני, תחתון, שמאלי

border-style:dashed solid groove inset;

או הגדרת העיצוב במספר שורות:

border-top-style:solid;

border-bottom-style:dotted;

border-left-style:groove;

border-right-style:inset;

## border-color

צבע הגבול, ערכי צבעים עבור הגבול

border-color:Red;

border-color:Red Blue Yellow Green;

## border-width

עובי סגנון הגבול

border-width:10px;  
border-width:10px 5px 15px 25px;



**איחוד כל התכונות**

ניתן לאחד את כל התכונות ביחד בשורה אחת לפי הסדר:

- border-style .1
- border-width .2
- border-color .3

border:solid 10px blue;

יש להוסיף רווח המפריד בין התכונות ולא פסיקים.



לכל אלמנט בדף HTML יש רקע(background) ולכל אחד מהם יש צבע.  
סגנונות הצבעים והרקע שולטים בתכונות אלה ומעצבים אותם לפי רצונך.

אלמנטים בנים יורשים בדרך כלל את הצבעים של הוריהם.  
לדוגמה אם הגדרת ל-body צבע רקע ירוק, יקבלו גם הכותרות <h1> צבע ירוק.

נהוג בדרך כלל לתת צבע רקע רק לדף אך ניתן לתת צבע רקע לאלמנטים מסוימים וכך להבליט אותם על רקע הדף כגון פסקאות וכותרות.  
תגית היוצרת ירידת שורה צובעת בהרקע את רוחב כל השורה לעומת תגית שלא יוצרת ירידת שורה שתצבע רק חלק ממנה.

**background-color**

צבע רקע לאלמנט:

background-color:Gray;

## background-image

+

מקבלת ערך URL של תמונת רקע כלשהי, או את המילה none.  
תמונת רקע יכולה לבוא על כל הדף או מאחורי אלמנט מסוים.  
בכל מקרה בו התמונה קטנה מגודל האלמנט היא תכפיל את עצמה  
מספר פעמים עד שתמלא את כל האלמנט,  
אם התמונה גדולה מדי לגודל האלמנט היא תיחתך.

-

```
background-image:URL('ballon1.jpg');
```

## background-repeat

מגדירה לדפדפן האם ואיך להכפיל את תמונת הרקע באלמנט,  
כאשר זה האחרון גדול מהתמונה. התכונה מקבלת אחד מהערכים:  
repeat-x, repeat-y, no-repeat , repeat-x

repeat-x – תכפיל את התמונה לרוחב הדף, לרוחב השורה העליונה  
repeat-y – תכפיל את התמונה לגובה הדף, לרוחב הפינה השמאלית  
no-repeat – התמונה לא תוכפל ותמוקם בפינה השמאלית העליונה.  
Repeat – ברירת המחדל, התמונה תשוכפל עד שתמלא את כל גבול  
האלמנט.

לדוגמא:

```
background-repeat:no-repeat;
```

## background-position

מגדירה לדפדפן היכן למקם את תמונת הרקע.  
ברירת המחדל היא הפינה השמאלית העליונה של האלמנט. תכונה זו  
יכולה לקבל ערך אחד או שניים.  
אם רק ערך אחד ניתן, הערך ישמש את שתי התכונות (מיקום אופקי  
ואנכי).  
אם ניתנו שני ערכים הראשון מתייחס לאופקי והשני לאנכי. לדוגמא:

```
background-position:center bottom;
```

ניתן גם לקבוע מרחק אופקי ואנכי בערכים של פיקסלים ושל אחוזים.  
דוגמא:

```
background-position:45% 50%;
```

## background-attachment

+

היא תכונה הפועלת המאפשרת למקם את תמונת הרקע כקבועה (fixed) או נעה (scroll) בגלילה.  
דוגמא:

-

background-attachment:fixed;

כאשר נגלול את המסמך, תמונת הרקע תישאר קבועה במסך במיקום שהוגדר לה באופן יחסי לחלק שבו אנו נמצאים במסך דרך מאפיין ה background-position.  
דוגמא:

background-attachment:scroll;

ערך זה הינו ערך ברירת המחדל, תמונת הרקע תישאר קבועה במיקום שהוגדר לה ולא תשנה את מיקומה היחסי כאשר נגלול את המסמך.

## קיצור הגדרות תכונות הרקע

התכונה background מאגדת את כל תכונות הרקע ומאפשרת לקצר הגדרות.  
אין חוקים לגבי סדר הערכים ואפשר לשים אותם בכל סדר שהוא, מופרדים זה מזה ברווח:

background:URL('ballon1.jpg') repeat-x center scroll;

### עיצוב רשימות

באמצעות גיליונות סגנון ניתן לעצב את המראה של פריטים ברשימות. דפדפנים מתייחסים לרשימות כמו כל בלוק אחר של טקסט, פרט לעובדה שלבלוק זה יש מקדם מסוג מסוים, כגון מספור (במקרה של OL) או סימון/תבליט (במקרה של UL).  
את צורת ומיקום הסימון ניתן לעצב.

## list-style-type

סגנון התבליט או המספור שיוצג לפי כל סעיף ברשימה.  
משמש לשני תפקידים:

## 1. סגנונות תבליט:

circle ○

square ○

disc ○

ברירת המחדל היא disc.

## 2. סגנונות מספור:

decimal (מספור רגיל) ○

lower-roman, upper-roman (אותיות קטנות וגדולות

ביוונית i)

lower-alpha (אותיות קטנות באנגלית) ○

upper-alpha (אותיות גדולות באנגלית) ○

ברירת המחדל היא decimal.

none – הצגת הפריטים ללא סגנון מספור או תבליט.

## list-style-image

התכונה list-style-image מקבלת URL של תמונה כלשהי או את המילה none.

אם יש לדפדפן גישה אל התמונה הוא יציג אותה, אם לא, יציג הדפדפן את מה שהוגדר כ- list-style-type.

כדי לקבל תוצאות יפות כאן מומלץ להשתמש בתמונות קטנות. לדוגמא:

```
list-style-image:URL('After Long Run c2.lco');
```

## list-style-position

מיקום התבליט או המספור בהתאם למיקום אשר נקבע לו על ידי הדפדפן.

ניתן להגדיר כיצד תת הסעיף של התבליט יופיע האם תבוצע הזחה פנימה או החוצה.

כדי להבין זאת בצורה הטובה ביותר צרו רשימה עם מספר רמות של סעיפים וקיבעו הזחות שונות עבור כל רמה

```
list-style-position:inside;
```

```
list-style-position:outside;
```

## כתיבת התכונות בקיצור



ניתן לכתוב את כל התכונות בקיצור:



list-style:square URL('After Long Run c2.lco') outside;

אין חשיבות לסדר כתיבת הערכים.

Visibility

כל אלמנט ב body יכול להיות גלוי או מוסתר.

התכונה visibility מאפשרת למתכנת להעלים מהעין דברים מסוימים שלא רוצים שהמשתמש יראה.  
ערכים אפשריים הם hidden וכן visible  
הדפדפן שומר עבורם מקום אבל לא מציג אותם בדף.

```
#subItem
{
visibility:hidden;
}
```

התכונה שימושית מאוד כאשר עובדים עם DHTML.

Display

תכונה דומה אחרת היא display, היכולה לקבל את הערכים block, inline, list-item וכן none.  
התכונה display:none מעלימה את האובייקט, ולא שומרת לו מקום (להבדיל מ- visibility).

לדוגמה:

```
display:none;
display:block;
```

שימוש בערכים block,inline מאפשרת להציג מחדש את האלמנט שהוסתר.

- Block – מגדירה שהאלמנט יתנהג כ block-element
- Inline – מגדירה שהאלמנט יתנהג כ Inline-element

בשימוש ב DHTML ניתן ליישם את השימוש בתכונה באופן מרבי.



הגדרת עיצוב המגדירה לאלמנט שעוטף אלמנטים נוספים בתוכו איך להתנהג במידה והאלמנט הראשי קטן בגודלו מהאלמנטים המקוננים שבו.

לדוגמא יצרנו דיב עם הגדרות של גובה ורוחב והוספנו לתוכו תגית <img> שמימדיה גדולים מגודל הדיב. היות והתמונה גדולה מגודל הדיב התמונה תגלוש מחוץ לדיב.

דבר זה יכול להיות בעייתי במידה ומיקמנו אלמנטים נוספים מסביב לאותו דיב ואותם אלמנטים יזוזו לצדדים. בבת אחת כל האתר שלנו משתנה פסקאות ירדו שורה, Divs יסתדרו אחד מתחת לשני או ידחפו מחוץ למסך ויופיע סרגל גלילה אנכי.

ניתן להגדיר עבור הדיב כיצד יתנהג במצב שגודלו קטן מדי

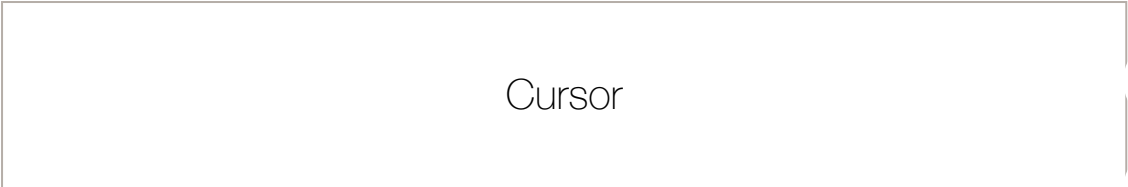
overflow:hidden;

כל התוכן החורג מהדיב ייחתך , במצב זה תוצג רק חלק מהתמונה והדיב יישאר באותו הגודל. לא היה ניתן לראות את המשך אלמנט.

דוגמא נוספת:

overflow:scroll;

בדוגמא השנייה יופיעו סרגלי גלילה והדיב לא ישנה את גודלו.ניתן יהיה לגלול בתוך הדיב ולראות את כל האלמנט. התכונה שימושית גם במקרה והגדרנו עבור דיב גובה מסוים ורשמנו בתוכו תגיות <p> שתופסות יותר גובה מגובה הדיב הקיים.



סמן העכבר מקבל מראה שונה כאשר מעבירים אותו מעל אלמנטים שונים  
(נראה כמו יד במעבר על קישורים, וכמו שעה במעבר על שדות להכנסת טקסט, למשל).  
ניתן לשנות את מראה העכבר גם בעזרת תכונת cursor, שיכולה לקבל את הערכים הבאים:

- default •
- pointer •
- move •
- e-resize •
- ne-resize •
- nw-resize •
- n-resize •
- se-resize •
- sw-resize •
- s-resize •
- w-resize •
- text •
- wait •
- help •
- hand crosshair •
- auto •

לדוגמא:

cursor:help;

ניתן למקד את תשומת ליבו של המשתמש לחלקים מסוימים באתר כגון קישור עבור למיקום מסוים באתר.



## Zoom



תכונת ה- zoom יכולה לקבל את הערכים: normal, מספר או אחוזים.  
היא פעילה רק בחלק מהדפדפנים.

```
p:hover  
{  
  zoom:200%;  
}
```

בעת מעבר העכבר מעל פסקה גודל הכתב יגדל וניתן יהיה לקרוא את  
הכיתוב בצורה נוחה יותר.

## Opacity

### הגדרת סגנון שקיפות

firefox משתמש במאפיין

```
opacity:x;
```

בעוד שדפדפן IE משתמש במאפיין

```
filter:alpha(opacity=x);
```

### ערכים אפשריים:

Fire fox – ערכים נעים בין 0-1

IE – ערכים נעים בין 0-100

ככל שהערך קטן יותר כך רמת השקיפות של האלמנט גבוהה יותר.  
לדוגמא:

opacity:0.4;  
filter:alpha(opacity=40);

תרגילים

תרגיל 1

צרו את דף ה Html הבא ועצבו אותו לפי ההגדרות הרשומות למטה  
בעזרת Inline CSS.

Welcome to our SuperCar Web Site

We sell new and used cars of all kinds

ave fast cars , s l o w cars , blue cars and small cars and all of them are good cars

We have been in business since 1976  
We own 12 agencies around the country

We welcome you to our site and hope you find a car you like here

שורה 1 – תגית h1

עיצוב – טקסט ממורכז , מודגש , קו תחתי , צבע כתב לבחירתכם

שורה 2 – תגית P

עיצוב – טקסט ממורכז , סגנון גופן לבחירתכם , גודל גופן 20 פיקסלים,  
צבע כתב לבחירתכם

שורה 3 – תגית span שבתוכה קיימת תגית span

עיצוב – לפי הדוגמא , בנוסף לשנות צבע כתב

שורות 4+5 – תגית פסקה עם תגית br

יישור טקסט לשמאל , גודל כתב 25 פיקסלים , צבע כתב לבחירתכם

## שורה 6 – תגית <div>

עיצוב – יישור טקסט למרכז

כמו כן ניתן להוסיף סגנונות עיצוב משלכם לתגית <div> ועוד הגדרות עיצוב לפי שיקולכם.

## תרגיל 2

צרו דף HTML עם התגיות הבאות:

1. תגית <h1> בראש הדף המכילה את שם האתר
2. תגית <h2> עם נושא ראשון שעליו תדברו באתר
3. מתחת לכותרת <h2> יש ליצור פסקה המתארת את הנושא בעלת 3 שורות
4. יש לחזור על שלבים 2,3 פעמיים נוספות
5. הגדרות עיצוב:

- תגית <div> - כיוון כתיבה מימין לשמאל , צבע רקע לבחירתכם.
- תגית <h1> – מיושרת למרכז , קו תחתי , נטוי , מודגש , סגנון גופן וצבע כתב לבחירתכם.
- תגיות <h2> – מיושרת לימין , קו תחתי צבע גופן וסגנון גופן לבחירתכם
- <p> – מיושרת לשמאל , גודל כתב 2em , סגנון כתב וצבע כתב לבחירתכם , הכנסת השורה הראשונה של כל פסקה.

## תרגיל 3

פתחו את הקובץ <cssEx2.html> והגדירו את הגדרות העיצוב הבאות:  
– <Body>

- צבע רקע לבחירתכם ותמונת רקע לבחירתכם

- <Div>

- לכל אחד מה <Divs> יש להגדיר כיוון כתיבה מתאימה

-<h1>

- טקסט ממורכז, נטוי , צבע כתב לבחירתכם , גודל וסגנון כתב לבחירתכם.



– P



- גודל צבע וסגנון כתב לבחירתכם

– Span

- צבע כתב שונה מתגית p
- גודל כתב גדול ב 10 פיקסלים מאשר תגית ה p

- hr

- רוחב קו 50 אחוז
- גובה קו 5 פיקסלים
- צבע קו לבחירתכם

## תרגיל 4

צרו את הרשימה הבאה בלחיצה על שם הסרט תופנו לאתר בו תוכלו לקרוא מידע על הסרט הספציפי.  
מעל לרשימה הוסיפו כותרת h1 ובה רשום שם האתר, עיצוב הכותרת כרצונכם.  
יש לעצב את רמות הרשימה בעיצוב לפי בחירתכם – צבע, סגנון כתב , גודל וכו'.

- סרטי אקשן
  - הפריצה לאלקטרז
  - מטריקס
  - שליחות קטלנית 3
- סרטי קומדיה
  - גברים בשחור
  - לילה מטורף במוזיאון
  - אייס ונטורה
- סרטי אנימציה
  - שרק

- רטטוי
- עידן הקרח
- סרטי דרמה
- שהארי פגש את סאלי
- טיטאניק
- פורסט גאמפ

## תרגיל 5

צרו דף html עם כותרת h1 שבה כתוב – the playoff is coming  
הכותרת מיושרת למרכז ומתחתיה תגית hr בעלת רוחב של 50%,  
גובה 5 פיקסלים וצבע קו לבחירתכם.  
מתחת יש להוסיף תגית marquee עם תמונה.  
צרו רשימת תבליטים עם 3 רמות:

- מערב
- סאן אנטוניו
- A.

טים דאנקן
- B.

טוני פארקר
- C.

מנו ג'ובילי
- D.

לוס אנג'לס לייקर्स
- E.

קובי בריאנט
- F.

פאו גאסול
- G.

למאר אודום
- מזרח
- בוסטון סלטיקס
- A.

קוין גארנט
- B.

פול פירס
- C.

ריי אלן
- D.

אורלנדו מג'יק
- E.

דוויט הווארד
- F.

רשארד לואיס
- G.

מייקל פיאטרוס



שמות השחקנים ברמה ה - 3 הם למעשה קישורים לאתר של הקבוצה.  
יש ליצור עיצוב עבור תגיות Li לפי מיקומן ולא בעזרת מחלקות



- רמה 1- תבליט מסוג עיגול , צבע כתב לבחירתכם, טקסט מודגש, נטוי
- רמה 2 - תבליט מסוג ריבוע , צבע כתב וגודל כתב לבחירתכם
- רמה 3 - תבליט מסוג upper-alpha

בנוסף יש להגדיר עיצוב לרקע של הדף מסוג תמונה  
יש להגדיר עיצוב עבור תתי אלמנטים של תגית <a> בנוסף יש להגדיר  
צבע רקע ותמונת רקע לדף.

## מודל הקופסא

### מודל הקופסא

CSS מגדירה מודל של קופסא אשר מוקפת בכמה אזורים שונים.  
מודל הקופסא מגדיר כיצד יוצג כל אלמנט, כל אלמנט בדף נחשב למלבן  
של האלמנט המכיל את תוכנו ומכיל גבול, שוליים ומרחק מהגבול  
(padding)  
באמצעות CSS ניתן להגדיר גם את גבולות באלמנט.



Padding

המרחק בין תוכן האלמנט לבין הגבול של אותו אלמנט.  
לכל אלמנט יש אזור שמקיף את תוכנו והוא הרווח הקיים שנותר לה עד לגבול האלמנט,  
במידה ונוסיף לאלמנט מסוים צבע רקע (background-color) נוכל לראותו.

דוגמא: הגדרת המרחק מקצהו השמאלי של גבול האלמנט

```
padding-left:10px;
```

דוגמא: הגדרת ארבעת המרחקים מהעליון לפי כיוון השעון של האלמנט מהגבולות שלו.

```
padding:20px 10px 30px 20px;
```

Border

גבול האלמנט, ניתן להציג לכל אלמנט את הגבול שלו, ניתן להשפיע על סגנון הגבול, עובי הגבול ועוד.

Margin



המרחק בין גבול האלמנט (border) לגבול של האלמנט הבא במסמך השוליים למעשה יוצרים את הרווח בין האלמנטים השונים בדף.

Padding, margin, border ערכי ברירת המחדל שלהם הינם אפס. אבל לפעמים קיימות הגדרות עיצוב ברירת מחדל של הדפדפן ולכן ניתן להגדיר לכל האלמנטים ערכי 0. ניתן להגדיר לשוליים ערך שלילי.

האלמנטים הינם שקופים ולא רואים אותם מסביב לאלמנט וניתן להגדירם עבור כל צד של האלמנט בנפרד – עליון, שמאלי, תחתון וימני. ב CSS height מתייחס לגובה הכולל של אלמנט, width מתייחס לרוחב הכולל של אלמנט הכולל margin, padding

לדוגמא:

```
#top
{
margin:10px;
padding:5px;
width:70;
}
```

במידה ויצרנו שוליים בגודל 10 פיקסלים ו Padding באורך 5 פיקסלים ונרצה שרוחב האלמנט הכולל יהיה 100 פיקסלים נצטרך להגדיר רוחב של 70 פיקסלים.

ניתן להגדיר כל צד בנפרד:

```
margin-left:10px;
```

או להגדיר בבת אחת שוליים עבור כל הצדדים:

```
margin:30px 20px 10px 15px;
```

במידה ונמקם שתי פסקאות אחת מתחת לשנייה אז השוליים הגדולים יותר יקבעו את המרחק בין הפסקאות.



ניתן גם להגדיר גודל שוליים אוטומטי. גודל שוליים 0 קובע שלא תהיה הפרדה בין אלמנט זה לאלמנט הבא במסמך ,



auto



auto מגדיר ריווח אחיד מימין ומשמאל ביחס לרוחב האלמנט העוטף אותו.

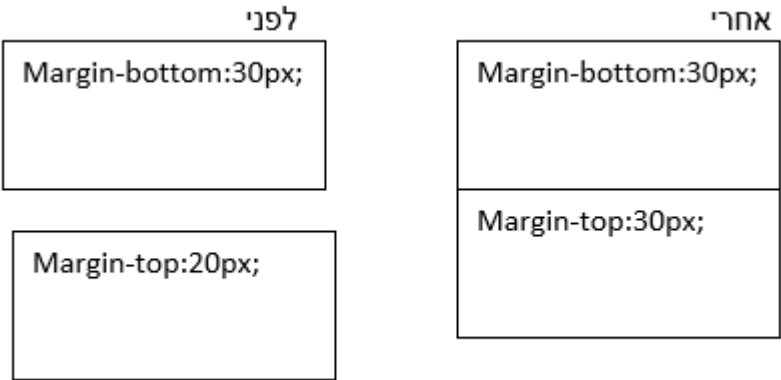
```
p
{
margin:0 auto;
width:300px;
}
```

רוחב הפסקה 300 פיקסלים והפסקה מיושרת במרכז השורה יחסית לרוחב המסך.  
השוליים הימניים והשמאליים שווים בגודלם.  
בדרך זאת נוח למקם את הדיב הראשי של הדף וכך להבטיח שכל תוכן האלמנטים יוצגו תמיד באמצע המסך.

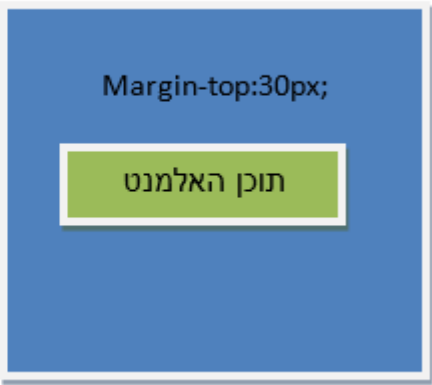
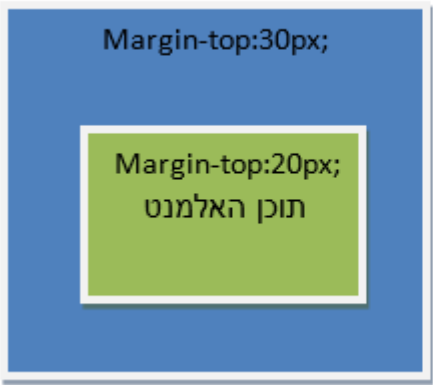
התנגשות שוליים

כאשר שני אלמנטים אנכיים מתנגשים הם ייצרו שוליים מאוחדים , גובה השוליים החדשים יהיה שווה לגובה השוליים הגדולים מבין שני האלמנטים

לדוגמא:



כאשר אלמנטים מקוניים בתוך אלמנט ויש התנגשות שוליים אז השוליים של האלמנט המקונן ואלמנט האב יאוחדו.



השוליים העליונים כעת הינם 30 פיקסלים.

ניתן לפתור את הבעיה של התנגשות שוליים של אלמנט פנימי עם האלמנט העוטף אותו בעזרת הוספת `padding:1px` לאלמנט החיצוני, התוספת מרווח של פיקסל אחד לרוב מפרידה של השוליים של האלמנטים השונים ומונעת "התנגשות".

שוליים של אלמנטים צפים (float) או מיקום אבסולוטי (position) לא מתנגשים.

תרגילים

תרגיל 1

- צרו פסקה המכילה שתי שורות והעתיקו את הפסקה שיצרתם כך שתופיע 20 פעם באתר
- יש ליצור 4 מחלקות עיצוב ולהחיל עיצוב של כל מחלקה על 5 פסקאות באתר.
- העיצובים עצמן אמורים לכלול לפחות חלק מההגדרות הבאות :
  - יישור טקסט
  - טקסט מודגש
  - טקסט נטוי

○ צבע כתב

○ כניסת שורה ראשונה של פסקה

○ סגנון גופן

○ גודל גופן

○ צבע רקע

○ סגנון גבול

○ צבע ועובי גבול

○ Width

○ Padding

○ margin

○ cursor

- בנוסף ניתן להוסיף הגדרות עיצוב משלכם.
- יש ליצור גם סגנון עיצוב במעבר עכבר מעל לתגית `h`
- ניתן להגדיר עבור כל מחלקה סגנון תת אלמנט `hover` משלה.

## תרגיל 2

- צרו רשימת הגדרות הכוללת לפחות 5 מונחים, עבור כל מונח יש לפרט לפחות 3 שורות, ניתן להעזר באתרי האינטרנט ולהעתיק משם את המידע הדרוש.
- בסוף כל תאור יש ליצור קישור לאתר שממש נלקח התיאור להמשך קריאה.
- בראש האתר יש ליצור רשימת קישורים שכל קישור מהווה עוגן לכל אחד מהמונחים שברשימה.
- הגדרות עיצוב – יש להגדיר הגדרות עיצוב כלליות עבור התגיות – `dd,dt,a`
- יש ליצור שתי מחלקות עיצוב ולהחיל אותן על שתי תגיות `dd` לבחירתכם
- יש ליצור עיצוב עבור תתי האלמנטים של `<a>`
- יש ליצור סגנון עיצוב של מעבר עכבר מעל לתגית `dd,dt`.

## תרגיל 3

פתחו את הקובץ `cssEx4.html`

הגדירו שבלחיצה על שם המשחק המשתמש יועבר לאתר בו ניתן

לרכוש או לקרוא על המשחק.

+

הוסיפו לדף תגיות <div> כדי ליצור חלוקה והגדירו עבור כל דיב עיצוב רצוי.

-

יש להגדיר עיצוב לדף עבור כל התגיות. ניתן להשתמש ב class , תתי אלמנטים, id, עיצוב לפי מיקום תגית וכו'

## טיפוסי תגיות

טיפוסי תגיות

### שני הטיפוסים של התגיות הן:

- inline elements
- block elements

### Inline Elements

תגיות ה-inline נועדו להיות באותה שורה עם אלמנטים נוספים ולא להתחיל איזור חדש. תגיות לדוגמא:

- <a>
- <span>
- <img>
- <input>

שנוסיף לדף את התגיות הללו הן יסתדרו אחת לצד השנייה שהרווח בין כל אלמנט לאלמנט ייקבע לפי הגדרת סגנון השוליים.

### Block Elements

תגיות ה-block מגדירות איזור שלם, איזור זה תופס את כל רוחב הדף ולכן יוצר ירידת שורה בסופו. יכולנו לראות את זה בצורה ברורה מאוד

כאשר ביקשנו להקיף את הכותרת בגבול.

תגיות לדוגמא:

+

- 
- <p> •
  - <ul> •
  - <ol> •
  - <li> •
  - <h1> •
  - <div> •

הצורה הכי קלה להבדיל ביניהם היא שאחת יורדת שורה בסופה, והשנייה לא.

CSS התאים את עצמו למודל זה וקבע כי חלק מהתכונות יחולו רק על inline-level וחלק רק על block-level.

ניתן לשנות את ברירת המחדל של האלמנטים (התנהגות האלמנט).

לדוגמא:

- תגית p שנרצה שלא תיצור ירידת שורה כמו inline box
- תגית span שמעוניינים שתיצור ירידת שורה כמו block element

< הצג דוגמא עם קוד

< נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      span
      {
        display:block;
      }
    </style>
  </head>
  <p>
```

```
{
  display:inline;
}
</style>
</head>
<body>
  <p>paragraph 1 </p>
  <p>paragraph 2 </p>
  <span>span 1</span>
  <span>span 2</span>
</body>
</html>
```

## התגיות <div> ו- <span>

לכל תגית html יש עיצוב משלה שהיא מחילה על העמוד. כלומר אפילו אם נפשיט את העמוד שלנו מכל CSS אפשרי ונוריד את כל התכונות האפשריות מתגיות ה- html עדיין נישאר עם עמוד בעל עיצוב מינימאלי, כגון ירידות שורה, מירכוזים, תמונות, פסקאות וכו'... לעיתים יש לנו צורך להחיל עיצוב של CSS אבל לוותר על כל הגדרה נוספת שאולי הייתה באה עם התגית. מצד שני CSS הוא עיצוב שחל רק על תגיות.

**יש לנו בעיה.**

**קיים צורך בתגית שלא מוסיפה אלמנט או משפיעה על אלמנט קיים אחר במסמך.**

לכן המציאו לנו 2 תגיות שזה בדיוק מה שהן עושות – כלום! (מבחינת תוכן).

אלה תגיות שכל מטרתן הוא לקבל עיצוב של CSS. תגית אחת לכל טיפוס תגיות של html.

**<span>**

היא תגית שמגיבה כמו inline level element.

## <div>

+

היא תגית שמגיבה כמו block level element.

-

ניתן להחיל עבורן את כל סגנונות העיצוב הקיימים ב CSS , מתנהגות כמו כל שאר תגיות ה html מבחינת קיבולן.

position

כאשר רוצים למקם אלמנט, משתמשים בתכונה position היכולה לקבל את הערכים absolute או relative. הערך absolute ימקם את האלמנט לפי הפרמטרים המצורפים, והאלמנטים הבאים אחר כך לא יתייחסו לאלמנט זה. לעומת זאת relative ימקם את האלמנט בהסטה של top ו-left . ושאר האלמנטים יושפעו מכך

## Relative

ימקם את האלמנט ביחס לאלמנטים אחרים בדף, האלמנט ימשיך לצוף בדף, במידה ונקבע מיקום אֶפֶס20;top, האלמנט ימוקם 20 פיקסלים מתחת לגבול האלמנט המקורי שלו כפי שמוקם בדף.

position:relative;

top:100px;

## absolute

ימקם אלמנט מסוים במקום מוגדר לפי הגדרות left, right, top, bottom, , שאר האלמנטים לא יתייחסו אליו ויתמקמו מבלי תלות בו. האלמנט ימוקם ביחס לאלמנט העוטף אותו ולא ביחס למיקום כתיבתו בדף.

position:absolute;

top:100px;

left:100px;

נניח ותגית ה body עוטפת את האלמנט , האלמנט תמיד ימוקם ביחס לדף מרחק של 100 פיקסלים מהקצה השמאלי ומהחלק העליון של הדף.  
(תחילת הדף מוגדרת תמיד כפינה השמאלית העליונה).  
ניתן לשים אלמנט אחד על אלמנט אחר ובאמצעות DHTML ליצור ארוע שיסתיר את אחד האלמנטים וכך למעשה ניתן ליצור תפריט נפתח.  
בתור ברירת מחדל האלמנט ימוקם בפינה השמאלית העליונה של האלמנט העוטף אותו,  
בדרך זו ניצב להציג אלמנט בכל מיקום רצוי בדף.

Fixed

ימקם את האובייקט כל הזמן במסך כאשר גוללים את הדף כלפי מטה.  
נניח שיצרנו דף html ומיקמנו בצידו השמאלי של הדף div אשר נרצה לשים בתוכו לוגו או טופס רישום לאתר,  
במידה ונקבע לו מיקום קבוע (fixed) הוא ישאר במיקומו הראשוני יחסית למרחקו מקצה הדף המוצג.  
לדוגמא:

```
#main
{
background-color:Fuchsia;
width:100px;
height:300px;
position:fixed;
}
```



כיוון שניתן להגדיר לאלמנטים מיקום קבוע יכול להיווצר מצב שבו אלמנט מסוים מכסה אלמנט אחר ומסתיר אותו,  
במידה ונרצה להגדיר מי מהאלמנטים יוגדר בשכבה העליונה ומי בתחתונה ניתן להשתמש בסגנון עיצוב z-index המקבל את הערכים 1,2,3 וכו'.  
מי שמקבל את הערך הגבוה ביותר יוצג בשכבה אחת מעל האלמנטים



לדוגמא: +

הצג דוגמא עם קוד

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

    <head>

        <style type="text/CSS">

            #main {
                width: 100px;
                height: 300px;
                background-color: Fuchsia;
                z-index: 1;
                position: absolute;
                left: 0px;
            }


            #first {
                width: 300px;
                height: 300px;
                background-color: Gray;
                z-index: 2;
                left: 100px;
                position: absolute;
            }


            #second {
                width: 300px;
                height: 300px;
                background-color: Blue;
                z-index: 3;
                left: 130px;
                position: absolute;
            }

        </style>

    </head>

    <body>
```

```
</style>

</head>

<body>

  <div id="main">

    <div id="first"></div>

    <div id="second"></div>

  </div>

</body>

</html>
```

+  
-

תרגילים

תרגיל 1

יש ליצור רשימת קישורים של לפחות 6 קישורים.  
מבלי להשתמש בתגית `<br>` יש ליצור הפרדת שורה עבור כל תגית  
קישור בעזרת סגנון העיצוב `display:block`.

יש להגדיר את הגדרות העיצוב הבאות עבור התגית:  
`Width, margin ,padding ,background ,border`

בנוסף יש להגדיר סגנונות עיצוב לתתי אלמנטים של תגית `<a>`

תרגיל 2

צרו 3 Divs בגדלים שונים מקמו את ה Divs כך שכל אחד מהם יעלה  
קצת על הדיב השני  
יש להעזר בפקודות `position ,left, right` ובאמצעות `z-index` לקבוע  
איזה דיב יהיה עליון

שליטה ומיקום שכבות בדף

## חסרונות בניית אתרים עם טבלאות



בעבר היה נהוג לבנות אתר עם טבלאות בלבד, הטבלאות היו משמשות לעימוד הדף ולסידור האלמנטים בצורה הרצויה. כיום נהוג לקבוע את עימוד הדף בעזרת CSS, להלן מספר סיבות לאי שימוש בטבלאות:

1. מערבב מידע הצגתי עם התוכן שלך.
2. גורם למשקל הקבצים של דפי האתר להיות גדולים שלא לצורך, מכיוון שהמשתמשים חייבים להוריד את המידע ההצגתי לכל דף בו הם מבקרים, במילים אחרות טעינה מהירה יותר של דפי האתר. גורם לתוספת עבודה ועלות עבור עיצוב מחודש של אתרים קיימים.
4. מקשה ביותר (ומייקר) לתחזק עיצוב עקבי בכל האתר.
5. דפים מבוססי טבלאות מקשים על הגישה לאינטרנט למשתמשים עם מוגבלויות ולגולשים דרך טלפונים ניידים ומכשירי כף-יד.

קיימות מספר שיטות ודרכים למקם שכבות בדף במיקום יחסי או מוחלט תוך שימוש בפקודות : position , margin , float ועוד פקודות נוספות.

כאן תוכלו לקרוא על חלק מהדרכים הקיימות

יישור עיצוב - שימוש ב position , margin

ברצוננו למקם דיב באמצע השורה , יצרנו דיב והגדרנו עבורו עיצוב לפי id

< הצג דוגמא עם קוד

## נסה אותי בתצוגה מלאה



```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #wrap
      {
        background-color:Gray;
        width:720px;
        height:300px;
        position:relative;
        left:50%;
      }
    </style>
  </head>
  <body>
    <div id="wrap"></div>
  </body>
</html>
```

נגדיר לדיב רוחב ונקבע לו מיקום יחסי (positive) ונגדיר לו מיקום  
50% מהקצה השמאלי,  
זה יצב את הקצה השמאלי של הדיב בחלקו האמצעי של הדף  
כעת כדי למקם את הדיב במרכז נגדיר שוליים שמאליים שליליים  
שמהווים חצי מרוחב הדיב שהוגדר

## הצג דוגמא עם קוד



## נסה אותי בתצוגה מלאה



```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #wrap
```

```
{
    background-color:Gray;
    width:720px;
    height:300px;
    position:relative;
    left:50%;
    margin-left:-360px;
}

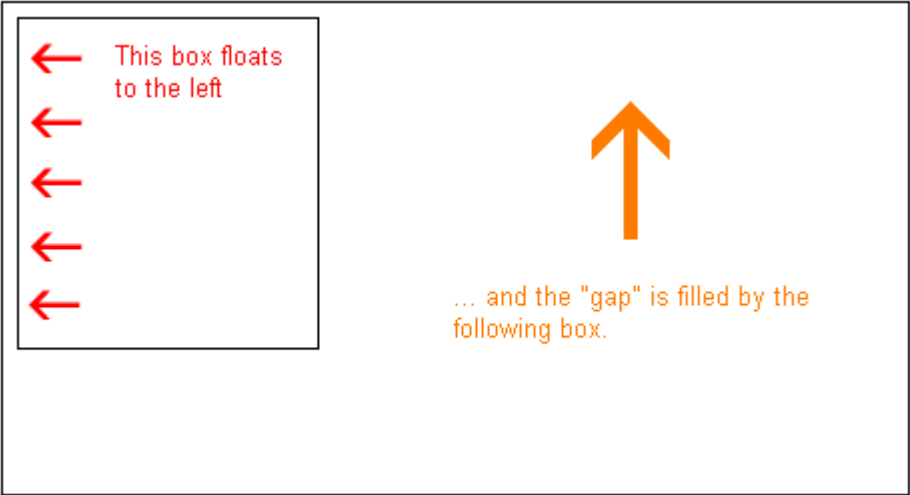
</style>
</head>
<body>
    <div id="wrap"></div>
</body>
</html>
```

הדיב יזוז שמאלה וימוקם בדיוק באמצע מבחינת רוחב הדף.

יישור Divs תוך שימוש בפקודת float

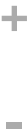
## אלמנטים צפים Float

אלמנט יכול להיות צף לימין או לשמאל באמצעות התכונה float. זה אומר שהתיבה עם התוכן שבתוכה צפים לימין או לשמאל המסמך. (או לאחד הצדדים של התיבה שמכילה אותם). לדוגמא נגדיר עבור תגית <img> שתצוף לשמאל אז מימין לתמונה הרווח יתמלא באלמנטים הבאים במסמך.



דוגמא:

float:left;



בעזרת תכונת ה float (ציפה) ניתן להגדיר חלוקת דף לאזורים שונים עם תגיות div וכך למעשה לשלוט על עימוד האתר בצורה קלה ונוחה מגדירים לדיב שרוצים שימוקם מצד שמאל float:left ואז עבור הדיב הבא בתור שרוצים שימוקם לצד הדיב הראשון מגדירים גם float:left לדיב האחרון בשורה שרוצים למקם ניתן להצמיד אותו לשמאל או לימין. תכונת ה - float למעשה מבטלת עבור תגית div את הגדרת block element.

**clear**

התכונה clear משמשת לשליטה כיצד תת אלמנטים של אלמנטים צפים במסמך יתנהגו. כברירת מחדל, תת האלמנטים זזים למעלה למלא את החלל שמתפנה כשתיבה צפה לצד. נראה את הדוגמה שלמעלה בה התיבה השנייה זזה אוטומטית לצד האלמנט שצף לשמאל. התכונה clear יכולה לקבל את הערכים right, left, both או none. העיקרון הוא, אם clear, נקבע ל-both עבור תיבה, השוליים העליונים של תיבה זו יהיו תמיד מתחת לשוליים התחתונים עבור תיבה צפה אפשרית שתגיע מלמעלה. האלמנט הבא יהיה בשורה שמתחת לאלמנט שקיבל את תכונת ה float. למעשה בוטלה אפשרות הציפה.

**דוגמא ראשונה:**

first	second	third

הצג דוגמא עם קוד &lt;

+

-

&lt; נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #first{
        width:30%;height:300px;
        background-color:Fuchsia;
        float:left;
      }

      #second{
        width:40%;height:300px;
        background-color:Gray;
        float:left;
      }

      #third{
        width:30%;height:300px;
        background-color:Blue;
        float:right;
      }
    </style>
  </head>
  <body>
    <div id="first"></div>
    <div id="second"></div>
    <div id="third"></div>
  </body>
</html>
```

לדיב הראשון הוגדר float:left, הדיב השני הושפע מכך ועלה למעלה  
למיקום הדיב הראשון

עבור הדיב השני הוגדר float:left מאותה סיבה של הדיב הראשון.  
עבור הדיב השלישי והאחרון בשורה הוגדר להיצמד לימין כך שהשורה  
לרוחב של 100 אחוז.  
  
שיטה נוספת: ההבדל הינו בהגדרת רוחב בפיקסלים ולא באחוזים.

+  
-

הצג דוגמא עם קוד

<

נסה אותי בתצוגה מלאה

<

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #first{
        width:250px;;height:300px;
        background-color:Fuchsia;
        float:left;
      }

      #second{
        width:250px;;height:300px;
        background-color:Gray;
        float:left;
      }

      #third{
        width:250px;height:300px;
        background-color:Blue;
        float:left;
      }
    </style>
  </head>
  <body>
    <div id="first"></div>
    <div id="second"></div>
```



```
<div id="third"></div>

</body>

</html>
```

הגדרנו לכולם float:left מכיוון שרוחב הדף כולו הינו מעל 900 פיקסלים, לכן אם היינו מגדירים לדיב השלישי float:right הוא היה נמצד לקצה הימני והיה נוצר רווח בין הדיב השני והשלישי. אם נוסיף תוכן לדיב שעולה על רוחבו הדיב עלול להתרחב ואז במצב שמספר Divs שממוקמים בשורה אחת ה Divs ירדו ויסתדרו אחד מתחת לשני ולא אחד ליד השני. לכן את הדיב האחרון בשורה נצמיד לימין ובמידה ורוחב הדיב הראשי יהיה רחב יותר יישאר לנו מרווח לבן קטן שמהווה מקום במידה ואחד ה Divs יגדל.

פתרון נוסף לבעיה הוא ליצור דיב ראשי לדוגמא id="main" שמגדירים עבורו רוחב מסוים וזהו הרוחב הכולל של כל ה Divs הפנימיים ואז לדיב האחרון בשורה נגדיר float:right כפי שמוצג בדוגמא הבאה.

### דוגמא שנייה:

first		
second	third	four
five		

main הינו דיב ראשי העוטף את שאר ה Divs שבתוכו. לכל דיב הוגדר רוחב float , לדיב האחרון הוגדר clear:both שמגדירה לו לא לצופ ליד Divs אחרים, אלא להתמקם בשורה משלו.

< **הצג דוגמא עם קוד**

< **נסה אותי בתצוגה מלאה**

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

+

```
<head>
```

```
<style type="text/CSS">
```

-

```
#main{
```

```
width:800px;
```

```
}
```

```
#first{
```

```
width:800px;
```

```
height:100px;
```

```
background-color:Fuchsia;
```

```
}
```

```
#second{
```

```
width:300px;
```

```
height:300px;
```

```
background-color:Gray;
```

```
float:left;
```

```
}
```

```
#third{
```

```
width:300px;
```

```
height:300px;
```

```
background-color:Blue;
```

```
float:left;
```

```
}
```

```
#four
```

```
{
```

```
background-color:Green;
```

```
width:200px;height:300px;
```

```
float:right;
```

```
}
```

```
#five
```

```
{
```

```
background-color:Yellow;
```

```
width:800;height:100px;
```

```
clear:both;
```

```
}
```

```
</style>
```

</head>

<body>

<div id="main">

<div id="first"></div>

<div id="second"></div>

<div id="third"></div>

<div id="four"></div>

<div id="five"></div>

</div>

</body>

</html>

דוגמא שלישית:

first		
second	third	four
	five	
six		

main הינו דיב ראשי העוטף את שאר ה Divs שבתוכו וממוקם באמצע מבחינת רוחב השורה בזכות הגדרת ה margin. לכל בדיב הוגדר רוחב ו float , לדיב האחרון הוגדר clear:both שמגדירה לו לא לצוף ליד Divs אחרים, אלא להתמקם בשורה משלו. לדיב החמישי הוגדר להיצמד לשמאל וככה הוא סוגר את הרווח שנשאר בגלל הגובה הקטן של הדיב השלישי.

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה <

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<style type="text/CSS">

#main{

https://t2k-pap-prod-assets.s3.eu-west-1.amazonaws.com/public/e6397bca-cda9-4986-86d9-992a34885a9c/b36c071d-e6a2-4901-8393-b390467170c1/t2k-u...

83/105

```
        width:800px;
        margin:0 auto;
    }
    #first{
        width:800px;
        height:100px;
        background-color:Fuchsia;
    }

    #second{
        width:300px;
        height:300px;
        background-color:Gray;
        float:left;
    }

    #third{
        width:300px;
        height:100px;
        background-color:Blue;
        float:left;
    }
    #four
    {
        background-color:Green;
        width:200px;
        height:300px;
        float:right;
    }

    #five
    {
        width:300px;
        height:200px;
        background-color:Red;
        float:left;
    }

    #six
```



```

    {
      background-color:Yellow;
      width:800;
      height:100px;
      clear:both;
    }
  </style>
</head>
<body>
  <div id="main">
    <div id="first"></div>
    <div id="second"></div>
    <div id="third"></div>
    <div id="four"></div>
    <div id="five"></div>
    <div id="six"></div>
  </div>
</body>
</html>
```

עבודה עם שכבות

כפי שראינו בדוגמאות הקודמות קיימות מספר דרכי עבודה שיוצרות שליטה על רוחב שכבות:

- Fixed •
- Fluid/liquid •
- Elastic •

Fixed

כאשר קובעים רוחב דיב בפיקסלים למעשה הגדרנו עבורו רוחב קבוע, דבר זה מקל על בונה האתר מכיוון שאם רוחב הדיב הינו 720 פיקסלים וברצוננו להוסיף תמונה נדע בוודאות שגודל התמונה צריך להיות 720 פיקסלים. Fixed הינו גודל קבוע שלא משתנה וקל להתאים עבורו אלמנטים

פנימיים.

השכבות נשארות קבועות אבל רזולוציית המסך משתנה בין משתמש למשתמש וקיים הבדל בין רזולוציה של 600\*800 לבין 768\*1024. יכול להיווצר סרגל גלילה אנכי מכיוון שרוחב ה Divs גדול מרוחב המסך ובנוסף מרכז ה Divs משתנה בהתאם לגודל המסך. לכן ניתן להשתמש בגודל באחוזים וליצור שכבות פלואידיות.

## Fluid/Liquid

נגדיר את רוחב ה Divs באחוזים, כך למעשה נתאים את גודל ה Divs למסך של המשתמש, הרוחב יגדל ויקטן בהתאם לרזולוציה. נגדיר גם margin, padding באחוזים על מנת שהמרווח ישתנה בהתאם למסך. ניתן גם לבצע שילוב בין סוגי שכבות , לדוגמא עבור הדיב הראשי להגדיר רוחב באחוזים ושאר ה Divs הפנימיים בערכי פיקסלים. דבר זה יכול להיות מאוד בעייתי מכיוון שאם גודל המסך קטן מגודל האלמנטים הפנימיים האלמנטים ישנו את מיקומם ויפיע סרגל גלילה אנכי. לכן מומלץ לשמור על אחידות בסוגי הערכים בין השכבות השונות.

## Elastic

יצירת רוחב דיב יחסי לגודל הכתב ולא יחסי לגודל המסך בדפדפן. נגדיר תחילה את גודל הכתב הבסיסי של הדף שיהיה שווה ל 10 פיקסלים, ברוב הדפדפנים הגדרת ברירת המחדל עבור גודל הכתב הינה 16 פיקסלים ולכן באחוזים ניתן לקבוע שגודל הכתב שווה ל 62.5% שזה למעשה 10 פיקסלים. לדוגמא

```
body
{
font-size:62.5%;
}
```

היות וכעת 1em שווה 10 פיקסלים, ניתן להגדיר את רוחב הדיב וגודל הכתב ביחידות של em. כל הערכים הינם באחוזים.

יצרנו למעשה שכבות אלסטיות שמשנות את גודלן ולא תלויות כלל ברזולוציית המסך.  
גודל השכבות משתנה וגם גודל הטקסט בהתאם.



## זליגה של שכבות

כאשר נקטין את גודל החלון עד נקודה מסוימת נראה שה Divs מתכווצים עד שהם יורדים שורה מפאת חוסר מקום, וזאת בגלל שרוחב הדף קטן יותר מרוחב מספר ה Divs בשורה. ניתן לפתור בעיה זאת בעזרת שימוש בתכונה max-width המגדירה שמגודל מסך מסוים לא תהיה השפעה על מיקום האלמנטים בדף והם ייחתכו מהמסך אבל לא יפריעו לעימוד הדף. לדוגמא הנחנו שני Divs באותה שורה ללא תכונת max-width כאשר היינו מקטינים את הדף מתחת לגודל מסוים ה Divs היו מסתדרים אחד מתחת לשני. בעזרת תכונת ה Max-width ה Divs יישארו באותה שורה אבל חלק מהדיב יחתך ולא יופיע על המסך, יופיע סרגל גלילה אנכי.

```
#main
{
width:72em;
text-align:left;
margin:0 auto;
max-width:75em;
}
```

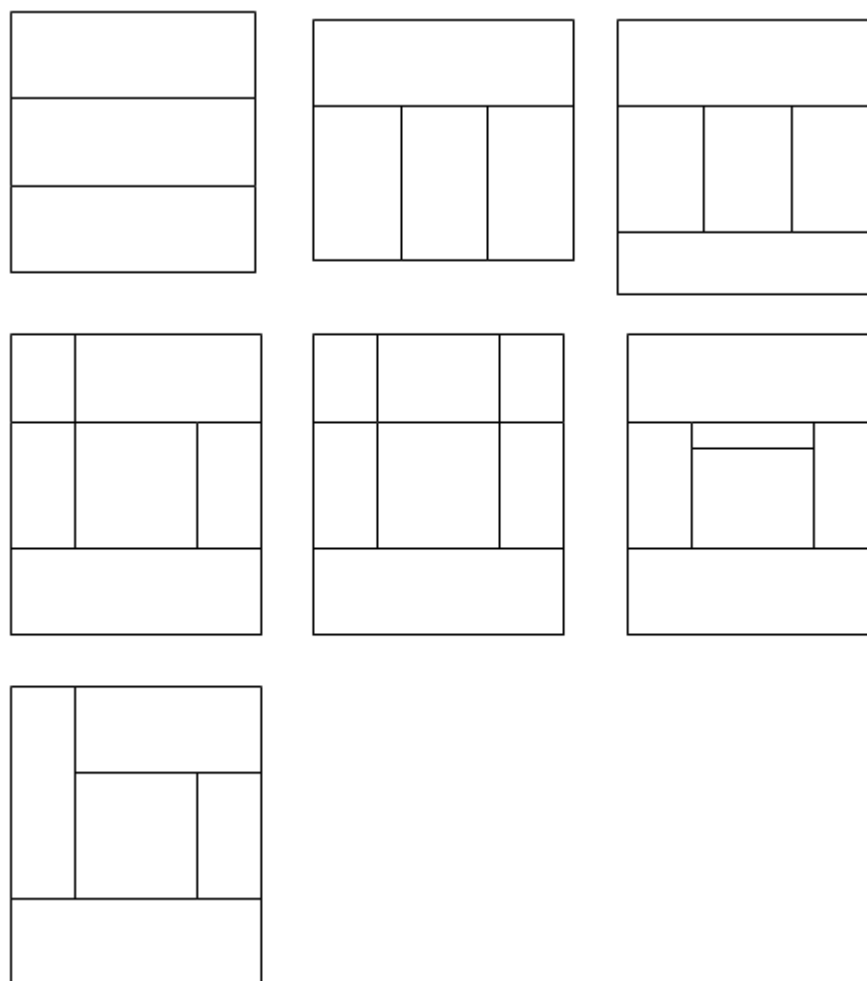
## תרגילים

### תרגיל 1

בניית שלד אתר מ Divs – קביעת מיקום עבור Divs במסך בכל אחד מהסעיפים הבאים יש לקבוע שסדר ה Divs יהיה לפי המפורט

## בתרגיל

עבור כל דף קיים דיב ראשי העוטף את שאר ה Divs הפנימיים, יש לתת הגדרות גובה, רוחב וצבע רקע לכל ה Divs.  
(לא להגדיר צבע רקע לדיב הראשי).  
לכל תגית div יש להגדיר id וליצור עיצוב עבור הדף פעם אחת ב internal CSS ופעם שנייה בקובץ חיצוני (external CSS).  
הדוגמאות הינן להמחשה בלבד , ניתן לקבוע לכל דיב איזו מידת גובה ורוחב שנראית לכם הכי נוחה עבור שלד אתר אופציונאלי עבורכם.  
סדר הסעיפים בתרגיל הינו מהדיב השמאלי העליון לכיוון ימין.



## תרגיל 2

פתחו את הקובץ **cssEx3.html**

- יש להגדיר ששמות המוצרים שבטבלה יהיו קישורים
- יש לעצב את העיצוב ב Internal/external CSS לפי ההגדרות הבאות.



- יש להגדיר עבור כל תגית <div> ערך id ולהגדיר עיצוב מתאים

לכל דיב

+

- יש להגדיר עיצוב עבור כל סוגי התגיות, העיצוב נתון לשיקול

דעתכם

-

### רשימת התגיות:

- Body,p,h1,hr,span,dd,dt,ul,li,ol,table,td,textarea,a

### יש להגדיר לפחות את העיצובים הבאים:

- עבור תגית a – יש להגדיר עיצוב עבור תתי אלמנטים
- עבור תגית a-הנמצאת בתוך הטבלה יש להגדיר עיצוב שונה מהקישורים שמחוץ לטבלה
- עבור תגיות – dd,dt להגדיר עיצוב בעת מעבר עכבר מעל לתגית
- Body – שוליים
- Li – מרווח גדול יותר בין ערך לערך ברשימה

שאר העיצובים כרצונכם.

## תרגיל 3

צרו טופס רישום הבנוי משני Divs המסודרים אחד ליד השני  
הגדירו לכל דיב רוחב מתאים

### מבנה הטופס:

יש להוסיף דיב עליון מעל לטופס הכולל תפריט קישורים לשאר דפי האתר, לעצב את הקישורים לפי רצונכם.

יצירת לוגו ותפריט ניווט

+

-

לוגו לאתר

ברצוננו להוסיף לאתר תמונת לוגו שתופיע בחלק העליון של הדף בקצה השמאלי או הימני העליון.  
ניתן להוסיף דיב לראש הדף אבל במידה ונוסיף בתוכו תמונה, אם התמונה קטנה מדי אז היא תשוכפל לרוחב ואורך כל הדיב, במידה והתמונה גדולה מדי היא תיחתך.  
עבור תמונת לוגו נבחר תמונה במימדים יחסית קטנים ונוסיף אותה כסגנון רקע עבור הדיב.  
בעזרת פקודת background-image.  
נגדיר שהתמונה תופיע פעם אחת ללא חזרות.  
במידה ונרצה להוסיף כיתוב ליד התמונה נוכל להגדיר Paddind עבור צד שמאל או ימין ואז למעשה הטקסט שנוסיף יתמקם אחרי התמונה ולא יסתירה.

לדוגמא:

עיצוב תגית הדיב

< הצג דוגמא עם קוד

< נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #top
      {
        background-color:Yellow;
```

```
width:800px;
height:100px;
background-image:URL('data:image/png;b
background-repeat:no-repeat;
background-position:left center;
}
</style>
</head>
<body>
<div id="top"></div>
</body>
</html>
```

## עיצוב תגית <h1>

אשר ממוקמת ראשונה בדף בתוך תגית הדיב שמכילה את תמונת הרקע.  
תכונת ה padding תאפשר שהטקסט יופיע מימין לתמונת הרקע.

```
h1{padding-left:30px;}
```

## דרך נוספת הינה להוסיף לתגית ה <body>

תמונת רקע ללא חזרות ולקבוע לתמונת הרקע הגדרת position מתאימה ועבור שאר התגיות להגדיר padding מתאים ביחס למיקומם בדף.

יצירת תפריט ניווט

אתר נוח למשתמש הינו אתר המכיל תפריט ניווט אחד או יותר המאפשר למשתמש לדפדף בין דפי האתר בצורה נוחה.  
את הקישורים בין הדפים השונים ניתן ליצור עם תגיות <a> ולכונן בתוכן תגיות נוספות ליצירת אפקט מרשים לדוגמא יצירת קישור מתמונה תוך שימוש בתגית <img>.  
כדי שכל הקישורים יהיו מסודרים בצורה נוחה ומרווחים ביניהם ניתן לחסוך את השימוש בתגית </ br>  
ולהשתמש בתגיות רשימה ולהגדיר עבורן סגנונות עיצוב ייחודיים.

תפריט אנכי

תפריט הממוקם בצידי האתר נשתמש בתגיות `<a>` , `<li>` , `<ol>` לדוגמא:

הצג דוגמא עם קוד <

נסה אותי בתצוגה מלאה <

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      ol {
        margin-left: -20px;
      }

      li {
        list-style-type: none;
        background-color: Gray;
        width: 100px;
        padding: 5px 0 5px 0;
        margin: 5px 0 5px 0;
        border: solid 2px purple;
      }

      li:hover {
        background-color: Transparent;
      }

      a {
        display: block;
      }
    </style>
```

</head>

<body>

<ol>

<li><a href="index.html">index</a></li>

<li><a href="gallery.html">gallery</a></li>

<li><a href="contact.html">contact us</a><

<li><a href="forum.html">forum</a></li>

<li><a href="aboutUs.html">about us</a></l

</ol>

</body>

</html>

הקישורים יופיעו במרחק מסוים מהאלמנט שלידם בזכות הגדרת ה -  
margin עבור תגית סוג הרשימה <ol>, לא יופיעו תבליטים עבור תגיות  
<li> ה  
וכל קישור יתפוס את כל הרוחב שהוגדר לתגית הרשימה בגלל הוספת  
התכונה display:block  
ניתן להוסיף תמונת רקע במקום צבע רקע או לשנות הגדרות של מיקום  
ושוליים לפי ראות עיניכם.

תפריט אופקי

תפריט הממוקם בצידי האתר  
נשתמש בתגיות <a> , <li> , <ol>  
לדוגמא:

<      הצג דוגמא עם קוד

<      נסה אותי בתצוגה מלאה

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<style type="text/CSS">

li {

https://t2k-pap-prod-assets.s3.eu-west-1.amazonaws.com/public/e6397bca-cda9-4986-86d9-992a34885a9c/b36c071d-e6a2-4901-8393-b390467170c1/t2k-u...

93/105

```

        list-style-type: none;
        background-color: Gray;
        width: 100px;
        padding: 5px 0 5px 0;
        margin: 5px 10px 5px 10px;
        border: solid 2px purple;
        float: left;
        text-align: center;
    }

    li:hover {
        background-color: Transparent;
    }

    a {
        display: block;
    }
</style>
</head>
<body>
    <ol>
        <li><a href="index.html">index</a></li>
        <li><a href="gallery.html">gallery</a></li>
        <li><a href="contact.html">contact us</a></li>
        <li><a href="forum.html">forum</a></li>
        <li><a href="aboutUs.html">about us</a></li>
    </ol>
</body>
</html>

```

בעזרת שינויים קטנים של הקוד הקודם ניתן כעת להשתמש בתפריט אופקי.

איך צורך להגדיר שוליים מהדף עבור תגית <ol>, מומלץ לעטוף את כל התפריט בדיב ולקבוע לו מיקום ביחס לדף. עבור תגיות <li> הוגדר float:left על מנת שיסתדרו אחת ליד השנייה באותה השורה וכמובן הגדרת padding כדי שתוכן האלמנטים לא יהיו צמוד אחד לשני.

הגדרת ה display:block נשארה כדי שנוכל ללחוץ בכל מקום ברוחב של תגית <li> ולהפעיל את הקישור הרצוי.

## תרגילים

### 1 תרגיל

יצירת אתר שהשלד שלו בנוי מ Divs, הדיב העליון מכיל תפריט ניווט, מתחתיו בצד שמאל יש למקם דיב עם תמונת רקע שתהווה את הלוגו של האתר. במידה וגוללים את המסך למטה הדיב יישאר קבוע במסך (יוצר מצב של לוגו רץ), יש להשתמש בהגדרה position כדי שהלוגו יישאר ממוקם במסך.

### 2 תרגיל

צרו תפריט ניווט אופקי עבור האתר, יש להשתמש בתגיות a, li יש לעטוף את כל התפריט בדיב ולקבוע עבורו רוחב ושוליים מהמדף יש להגדיר רוחב עבור תגיות li ולהגדיר שלא יופיעו תבליטים או מספור לצד כל קישור. בנוסף יש להגדיר padding ושאר העיצובים הנוספים לפי ראות עיניכם

### 3 תרגיל

צרו תפריט ניווט אנכי עבור האתר, יש להשתמש בתגיות a, li יש לעטוף את כל התפריט בדיב ולקבוע עבורו רוחב ושוליים מהמדף יש להגדיר רוחב עבור תגיות li ולהגדיר שלא יופיעו תבליטים או מספור לצד כל קישור. בנוסף יש להגדיר padding ושאר העיצובים הנוספים לפי ראות עיניכם

### 4 תרגיל

צרו דף באתר עם תפריט ניווט אופקי ואנכי, מקמו את התפריטים במסך בעזרת Divs. במעבר העכבר מעל לאחד מהקישורים בתפריט האופקי תופיע תמונת רקע שתעלם כאשר תעברו למקום לאלמנט אחר. בתפריט האנכי לאחר שביקרנו בקישור מסוים תופיע תמונה קטנה ליד הקישור

ובקישור שעדיין לא ביקרנו בו תופיע תמונה אחרת.

לדוגמא תמונה קטנה של ✓ , X

+

## תרגיל 5

-

כתבו בעזרת תגיות `<span>` בלבד דף באתר לפי ההנחיות הבאות,  
אסור לדף לכלול שום תגית `html` למעט תגיות `span`.

- על הדף לכלול את כותרת האתר ממורכזת בראש הדף
- מספר שורות המתארות את האתר, יש ליצור לפחות 6 שורות, כל שתי שורות מתארות צד מסוים באתר.
- כל נושא שצוין באתר יש לעצבו באופן שונה מבחינת צבע וסגנון כתב ועוד...
- יש ליצור מתחת לפירוט של האתר את התפריט ניווט האנכי באתר (לא ליצור קישורים לדפי האתר).
- יש ליצור עיצוב עבור התפריט ניווט וגם עיצוב של מעבר עכבר על רשימת הנושאים.

## תרגיל 6

צרו שלד של דף אינטרנט הבנוי מדיב עליון עם שורת ניווט באתר  
דיב ימני ושמאלי - קישורים, כתבות וכו'  
דיב אמצעי - תוכן הדף  
דיב תחתון - פרטים אודות בעל האתר  
את כל ה `Divs` עוטף דיב ראשי

העתיקו את מבנה האתר לשני דפים נוספים שבאתר  
רשמו בכל דף את כותרת האתר + לוגו החברה  
הוסיפו מספר שורות תוכן.

- דף ראשי - עמוד הבית
- דף שני - אודותינו, במה החברה עוסקת
- דף שלישי - טופס הצטרפות למועדון החברים

צרו קובץ הגדרות חיצוני המכיל הגדרות עיצוב עבור כל ה `Divs` ושאר התגיות.

לכל דיב יש להגדיר `id` שלפיו יוגדר סגנון עיצוב ספציפי.  
צרו שורת קישור או בצעו ייבוא לקובץ ה `CSS` בכל אחד מדפי האתר.



## תרגיל 7

+

צרו דף הבנוי מ Divs

-

דיב עליון שורת ניווט לאתר הבנויה מקישורים  
דיב שני - מכיל תיבת טקסט ולחצן submit שבלחיצה עליו המשתמש  
יועבר לאתר google ויופיעו תוצאות החיפוש לפי ערך החיפוש  
שהוקלד.

יש להוסיף תמונת רקע לדף

## תרגיל 8

צרו את מבנה הדף הבא לפי סדר התגיות הבא:

- חמישה קישורים לאתרים שונים לפי בחירתכם מתחת לקישורים
- תגית h1 ומתחתיה שתי תגיות p
- מתחת לתגית h2 שמתחתיה עוד שתי תגיות p
- מתחת לתגיות p יש ליצור טבלה של חמש שורות ושתי עמודות
- בעמודה הימנית בטבלה יופיעו קישורים לשמות סרטים ובעמודה השמאלית תופיע תמונה

סדר התגיות מימין לשמאל:

- תגית a כפול 5, p, h1 כפול 2, p, h2 כפול 2, table ובתוכה תגיות a + img.

יש להגדיר עיצוב לפי מיקום התגית עבור :

- תגית p הראשונה הממוקמת מתחת לתגית h2
- הגדרת עיצוב עבור כל הקישורים שבטבלה בלבד.
- הגדרת עיצובים עבור תתי אלמנטים של תגית a.

הגדרות העיצוב יכללו סגנונות עיצוב כרצונכם כגון :

- font, text, background, border, color, width, height, padding, margin

## CSS 3



## הקדמה

כיום כל הדפדפנים תומכים בגרסת CSS 2.1. גרסת CSS 3.0 נמצאת בפיתוח כבר מדצמבר 2005. למרות שהיא טרם אושרה רשמית, חלק מהדפדפנים המובילים תומכים כיום בכמה מהתכונות של CSS 3. ב-CSS 3 החליטו לחלק את השפה למודולים נפרדים, כך שכל מודול יוכל להתפתח באופן עצמאי ולא להיות תלוי באישור של המודולים האחרים. נכון למרץ 2017, ישנם קרוב ל-40 תקנים מאושרים. תקן זה צפוי לכלול אלמנטים גרפיים רבים כגון שינויי צבע וכיוון דינאמיים, צללים, שקיפות, פינות מעוגלות ועוד. כל מודול עוסק בהיבט אחר של השפה, לדוגמה המודול CSS Fonts עוסק בכל הנושא של פונטים, CSS Lists עוסק בתכונות של רשימות, וישנם מודולים כלליים יותר כמו CSS Math או CSS Positioning. כל מודול נמצא כיום בסטאטוס שונה של התפתחות. ניתן לעקוב אחר הסטאטוס של המודולים השונים באתר של ארגון התקינה הבינלאומי w3c:

<http://www.w3.org/Style/CSS/current-work.html>

ישנם עוד הרבה feature-ים חשובים נוספים שנמצאים כרגע בפיתוח, לדוגמה יצירה של Drop Shadows ל-Block-Level Elements (התכונה box-shadow), שימוש בתמונות כדי לצייר גבולות במקום בקווים פשוטים (התכונה border-image) ועוד. ניתן לדגום כמה מהתכונות החדשות של CSS 3 באתר הבא:

<http://www.css3.info>

להלן נפרט כמה מהמאפיינים החשובים שהתווספו ב-CSS 3.

## שקיפות



באמצעות התכונה opacity אפשר להפוך אלמנטים מסוימים בדף לשקופים באופן חלקי. התכונה opacity מקבלת ערכים בין 0 (שקוף לגמרי) לבין 1 (אטום לגמרי). התכונה הזו נתמכת בכל הדפדפנים חוץ מ-IE. עבור IE יש להגדיר תכונה מקבילה הנקראת filter שנותנת את אותו אפקט (תכונה זו מקבלת ערכים בין 0 ל-100). לדוגמא, נגרום לתמונה הבאה להיות חצי שקופה:

< הצג דוגמא עם קוד

< נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #img1
      {
        opacity: 0.5;
        filter: alpha(opacity=50);
      }
    </style>
  </head>
  <body>
    
  <head>
    <style type="text/CSS">
      #rounded {
        background-color: #BBBBBB;
        width: 250px;
        padding: 20px;
        border-radius: 10px;
        -moz-border-radius: 10px;
      }
    </style>
  </head>
  <body>
    <div id="rounded">This box should have rounded
  </body>
</html>
```

התכונה הזו נתמכת כיום בדפדפנים Firefox, Safari, Chrome, Opera ו-IE9. כיום, עבור התמיכה ב-Firefox יש להגדיר בנוסף את התכונה `moz--`

.border-radius

הקידומת moz מציינת שהתכונה מוכרת ספציפית רק על-ידי- Firefox (הקידומת של IE היא -ms- והקידומת של Safari היא -webkit-).  
ברגע שהתכונה border-radius תיתמך באופן מלא ב- Firefox, ניתן יהיה להשמיט את התכונה -moz-border-radius.

+

-

## טבלאות זברה

טבלאות זברה מאפשרות לצבוע כל שורה זוגית או אי-זוגית בטבלה בצבע רקע אחר.  
באופן הזה ניתן לשפר את הקריאות של טבלאות המכילות הרבה שורות.  
עד ל- CSS 3 השתמשו בטכניקות שונות ליצירת טבלאות זברה שדרשו כתיבת קוד לא מבוטלת (לדוגמא על-ידי הגדרת שתי מחלקות even ו- odd, ושיוך כל שורה בטבלה למחלקה אחרת).

ב- CSS 3 הוסיפו Selector בשם nth-child המאפשר לבחור ילדים של תגית מסוימת על-פי המיקום הסידורי שלהם תחת אותה התגית.  
הסלקטור הזה הוא די מתוחכם. הנוסחה הכללית שלו היא:

$:nth-child(an+b)$

המשמעות של הערך  $an+b$  היא שלכל ערך של  $n$  (שמתחיל מ- 0), יש לבחור את הילד של התגית הנמצא במקום ה-  $b + an$  תחת אותה תגית

(הקבוע  $a$  מציין את הדילוגים והקבוע  $b$  מציין את המיקום של הילד הראשון שיבחר).  
לדוגמא, כדי לבחור כל שורה זוגית בטבלה יש לרשום

$:nth-child(2n)$

כדי לבחור כל שורה שלישית בטבלה החל מהשורה השנייה יש לרשום

$:nth-child(3n + 2)$

וכד'.

עבור בחירה של שורות זוגיות או אי-זוגיות קיים גם הקיצור +

:nth-child(even) -

או

:nth-child(odd)

לדוגמא, נעצב את הטבלה הבאה על-ידי צביעת כל שורה אי-זוגית  
בטבלה בצבע רקע ירוק:

< הצג דוגמא עם קוד

< נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      table {
        margin: auto;
        width: 600px;
        text-align: left;
        border: 1px solid #99ff99;
      }
      tr:nth-child(2n+1) {
        background-color: #99ff99;
      }
    </style>
  </head>
  <body>
    <table>
      <tr>
        <td>first name:</td>
        <td>last name:</td>
      </tr>
```

```
<tr>
  <td>Dan</td>
  <td>Shaked</td>
</tr>
<tr>
  <td>Gil</td>
  <td>Dor</td>
</tr>
<tr>
  <td>Yuri</td>
  <td>Cohen</td>
</tr>
<tr>
  <td>Yoram</td>
  <td>Dayan</td>
</tr>
<tr>
  <td>Sam</td>
  <td>Gad</td>
</tr>
</table>
</body>
</html>
```



התכונה text-shadow מאפשרת להוסיף הצללות לכותרות, רשימות ופסקאות. תכונה זו מקבלת 4 ערכים:

- היסט אופקי - כמה רחוק מצד שמאל או מצד ימין לטקסט הצל צריך להתחיל. היסט שלילי מציין שה- shadow יתחיל משמאל לטקסט.
- היסט אנכי - כמה מתחת או מעל לטקסט הצל צריך להתחיל. היסט שלילי מציין שה- shadow יתחיל מעל לטקסט.
- מידת העמעום של ה-shadow (blurriness)
- צבע ההצללה

לדוגמא, נוסיף תגית <p> לדף ונגדיר לה אפקט של text-shadow:

< הצג דוגמא עם קוד

< נסה אותי בתצוגה מלאה

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style type="text/CSS">
      #shadow
      {
        font-family: Verdana, Geneva, sans-
        font-size: 24px;
        text-shadow: -4px 4px 3px #999;
      }
    </style>
  </head>
  <body>
    <p id="shadow">Text Shadow</p>
  </body>
</html>
```

תכונה זו נתמכת בכל הדפדפנים חוץ מ-IE.

נספח

תוספות לדפדפן ולינקים

נספח: תוספות לדפדפן



ניתן להוריד לדפדפנים כלי פיתוח

## Firefox

Web developers Add-ons for fire fox

לינק:

<https://addons.mozilla.org/en-US/firefox/addon/60>

מילת חיפוש :

Web developers tools

## IE

ניתן ללחוץ על מקש F12 החל מגרסת IE8 או לחפש כלי פיתוח באינטרנט.

## אתרים מומלצים

מדריכי לימוד CSS

[www.westciv.com/style\\_master/academy/CSS\\_tutorial/index.html](http://www.westciv.com/style_master/academy/CSS_tutorial/index.html)

אתר לדוגמא:

<http://www.csszengarden.com>

---

כל הזכויות שמורות לג'ון ברייס הדרכה בע"מ מקבוצת מטריקס