



מבוא לבנית אתרי אינטרנט והבנת המושג תגיות

נתחיל ללמוד איך לבנות דפי אינטרנט, לפני שאנחנו מתחילים בלמידת השפה המיועדת לכתובת דפי האינטרנט ננסה להבין מהי רשת האינטרנט, למה היא משמשת, למה היא מיועדת, נלמד מעט על סוגי האתרים הקיימים, ואיך רשת האינטרנט עובדת.

האינטרנט

האינטרנט היא רשת תקשורת כלל עולמית, שמהווה זירת התפתחות כלכלית ותרבותית רבת משמעות.

מקור השם 'אינטרנט' הוא קיצור המילים **Interconnected Network** המתאר קבוצת רשתות מחשבים המקושרת ביניהן.

האינטרנט מאפשר לאנשים לשתף מידע במהירות אחד עם השני, להשוות מחירים, להתעדכן בחדשות, לצרוך מוצרים ואף להקל על הבירוקרטיה מול משרדי הממשלה.

גורמים עסקיים רבים בחרו לפעול גם בזירת האינטרנט, בהמשך לפעילותם המסורתית, או בנפרד ממנה.

עמותות וארגונים רבים דואגים לאתר מרשים, שייצגם באינטרנט, בכדי שכתורמים או נזקקים פוטנציאלים שיידרשו לארגונם יוכלו לקבל זאת באופן המקיף והמהיר ביותר.

איך החל האינטרנט?

האינטרנט פותח בארה"ב בשנת 1969 לאחר שהרוסים שיגרו את לוויין הספוטניק הראשון לחלל על ידי ברית המועצות, ובארה"ב חששו ממתקפה גרעינית שתפגע בקווי התקשורת ותמנע יכולת תקשורת בתוך ארצות הברית ובין ארצות הברית לשאר העולם.

החיבור הראשון התבצע בין מחשב במעבדה של קליינרוק (ממציאי האינטרנט) לבין מחשב במכון SRI של אוניברסיטת סטנפורד. ובהדרגה, התחברו אליה מוסדות אקדמיה נוספים, כשהמטרה היתה שהמדענים בכל אחת מהן יוכלו להיכנס ולעבוד במחשבי האחרות על המחקר ההגנתי שעסקו בו במשותף.

רק ב-1994 פתחה ממשלת ארצות הברית את רשת האינטרנט האקדמאית לציבור הרחב והאינטרנט הפך להיות זה המוכר כיום.

המניע לבניית אתר ברשת

תחילה נבין מהי הסיבה לפיתוח אתר אינטרנט, קיימים מספר רב של סוגי אתרים, ביניהם אתרי חדשות, אתרי מסחר לשיווק ומכירת שירותים ומוצרים, אתרי תוכן למתן מידע לתחומים שונים: תחומי מדע, משפט, תחומים רפואיים (קופות חולים, בתי חולים, מרפאות וכדו'), תחומי תרבות, ממשל, בנקים וכו', פורמים להעברת ושיתוף מידע מקצועי וחברתי ועוד ועוד...

היום כל עסק וארגון המכבד את עצמו יבקש ייצוג באתר אינטרנט משלו, בין אם מדובר באתר מסחרי, ובין אם מדובר באתר תדמיתי טוב שישווק אותם בצורה האופטימלית ביותר. תכרות רבות אף מעניקות שירותי לקוחות באמצעות האינטרנט.

המעלה באתר אינטרנט היא נוחות וזמינות המידע שאפשר לתת לכל מתעניין, במשך כל שעות היממה ואפשרות להיחשף לקשת רחבה של לקוחות פוטנציאליים.

מהו אתר אינטרנט

בדומה לאוסף של דפים כרוך המכונה ספר או אוסף דפי פרסום ומידע המאוגד לעיתון, דפי האינטרנט מכילים תכנים רבים שונים ומגוונים, לרוב מלל, תמונות, סרטונים וכדו'. אוסף של דפי אינטרנט יחדיו מכונה אתר אינטרנט, דפי אינטרנט לרב מכילים קישורים לדפים אחרים, כך שנוצרת רשת כלל עולמית של דפי אינטרנט המקושרים, רשת האינטרנט הוא מרחב של מסמכי היפרטקסט - בעלי טקסטים מקושרים, ומכונה world wide web (ומכאן שמו www).

דפי אינטרנט בכל אתר, בין אם צופים באתר מבוסס מערכת ניהול תוכן, דוגמת וורדפרס, ג'ומלה, וויקס ו-דרופל ובין אם צופים באתר שנכתב רק על בסיס HTML טהור, בסופו של דבר תוכן אתרי האינטרנט מוצג בעזרת תכנה הנקראת דפדפן כדפי HTML.

איך עובד האינטרנט?

האינטרנט היא רשת תקשורת גלובלית המחברת בין רשתות מחשבים המאפשרת תקשורת ביניהם, כשכל אחד יכול להתחבר אליה באמצעות המחשב האישי שלו.

דפי האינטרנט יושבים על מחשב מרוחק ברשת, המחשב המרוחק נקרא בשם שרת, השרת הוא מחשב עם נפח גדול ומשרת לקוחות רבים, כל לקוח רוכש מחיצה על הדיסק הקשיח שלו ושם מניח את קבצי האתר שלו.

פרוטוקול HTTP

התקשורת בין המשתמש לאתר, מתבצעת באמצעות פרוטוקול תקשורת¹ שנקרא HTTP, פרוטוקול זה נוהל להעברת דפי HTML ואובייקטים שהם מכילים, (כמו תמונות, קבצי קול, קבצי וידאו וכדו').

התקשורת ב-HTTP מתחילה ביצירת שיחה בין הלקוח (הגולש במחשב) לשרת, ששם נמצא האתר שהלקוח מעוניין לצפות בו, ומתחילה כעין שיחה בין השרת למחשב הלקוח ונמשכת סדרה של בקשות ותשובות הלוך חזור שנשלחות על ידי הלקוח והשרת, בהתאמה. ניתן גם להכין קבצי HTML על מחשב מקומי ולהציג אותם באמצעות הדפדפן אך ורק במחשב ששם יושבים הקבצים, בקורס זה לא נעלה את התוכן שלנו לשרת מרוחק, אלא נבנה אתרי HTML ונציג אותם מהכונן המקומי.

פרוטוקול FTP

העברת קבצים מהמחשב לשרת באתר מתבצעת באמצעות פרוטוקול תקשורת המכונה ftp, פרוטוקול זה אחראי על העברת הקבצים ממחשבו של מנהל האתר למחשב המרוחק- השרת שבו האתר מאוחסן ברשת.

העברת הקבצים מתבצעת באמצעות תוכנת ftp כדוגמת FileZilla, תוכנה זו מאפשרת להתחבר לשרת, להוריד או להעלות לשרת קבצים. אופן העברת הקבצים מהמחשב לשרת

¹ (פרוטוקול תקשורת הכוונה לנוהל תקשורת. כלומר, אוסף של כללים המגדירים את אופן בקשת וקבלת נתונים במערכת תקשורת מסוימת וכולל כללים לייצוג המידע, איתות, אימות, ותיקון שגיאות לצורך העברת המידע בערוץ תקשורת. פרוטוקול מוכר ופשוט לכולנו הוא שיחת טלפון הכוללת כללים מוסכמים: הרמת השפופרת, קריאת "הלו", הצד מנגד עונה ב"שלום וברכה" ולאחר מכן יסביר את מהות ההתקשרות ותתחיל העברת המידע והשיחה. לפני ניתוק השיחה יפרדו האנשים ב"ביי" או "להתראות".

אולם ישנה גמישות, ואין בהכרח צורך בפרוטוקול קשיח ומוחלט, ולכן לא כל שיחת טלפון מתנהלת על-פי הפרוטוקול המדויק הנ"ל. אך כאשר מדובר ברשת תקשורת בין מחשבים, שימוש בפרוטוקולים מדויקים הכרחי על-מנת שהצדדים יבינו זה את זה ויכולו לספק שירותים זה לזה.

באמצעות תכנה זו מתבצע ביתר קלות, ומאפשר גרירה של מסמכים וקבצי מדיה מהמחשב לשרת ולהיפך.

כתובת IP

לכל מחשב בעולם כתובת IP מספרית המזהה אותו בין כל המחשבים, כתובת ה-IP מורכבת מארבעה מספרים כשערך כל מספר נע בין 0 ל-256 לדוגמא:

213.8.204.47

גם לכל אתר ברשת כתובת IP משלו, על מנת לגשת לאתר יש צורך ליצור חיבור לכתובת ה-IP של האתר, (כתובת ה-IP שבו השרת נמצא), לאחר מכן נשלחת בקשה הכוללת את הכתובת של האובייקט המבוקש (למשל דף מסוים באתר, כגון דף אודות וכדו') ופרטים נוספים על הבקשה ועל הלקוח (כמו כתובת ה-IP של הלקוח וכדו') השרת קורא את הבקשה, מפענח אותה, שולח ללקוח תשובה, לדוגמא מחזיר את הנתונים בכדי להציג את הדף ומנתק את החיבור ללקוח כשמסתיימת השליחה.

פרוטוקול ה-DNS

מאחר שקשה לזכור כתובות מספריות כדוגמת כתובות ה-IP, הומצא פרוטוקול המכונה פרוטוקול DNS, תפקידו להקל את השימוש של אנשים ברשתות התקשורת, ולתרגם את כתובות ה-IP לכתובות מילוליות, אותה קל יותר למשתמש לזכור, פרוטוקול ה-DNS מתרגם את הכתובת המילולית לכתובת ה-IP הנדרשת.

כאשר אנו רוצים לצפות באתר מסוים אנו כותבים את כתובתו המילולית בדפדפן, הכתובת מתורגמת לכתובת ה-IP של האתר, ובכך מתחברים לשרת ומקבלים ממנו את המידע - דפי האינטרנט בדפדפן לצפייה, דפי האינטרנט המוצגים בדפדפן הם תרגום קוד ה-HTML על ידי הדפדפן.

הדפדפן

דפי האינטרנט מוצגים באמצעות תכנה הנקראת דפדפן, המתרגמת את הקוד בו האתרים כתובים לתצוגה גרפית.

הדפדפן נקרא "צד לקוח", והשרת ששם מאחסנים את האתר נקרא "צד שרת".

ישנם סוגי דפדפנים שונים כרום (השייך לגוגל), אקספלורר (השייך למייקרוסופט), פיירפוקס, ספארי ועוד רבים.

שיטות לבניית אתר

ניתן לחלק את השיטות לבניית אתר לשני ענפים עיקריים:

האחד, בנייה הנעשית בעיקר על ידי מערכת ניהול תוכן כדוגמת וורדפרס, ג'ומלה ודרופל וכדו'.

והשנייה, בנייה הנעשית בעיקר על ידי קוד.

בכל מקרה לא משנה באיזו צורה אתר האינטרנט בנוי, כשנצפה באתר יוצגו לפנינו דפי HTML.

אתר סטטי לעומת אתר דינאמי

את אתרי הרשת ניתן לחלק חלוקה גסה לאתרים סטטיים ודינאמיים,

אתרים סטטיים הכוונה לאתרים שלא משנים את פניהם כלל, אתר סטטי הוא אתר בו הקוד, התוכן והפריסה אינם ניתנים לשינוי יומיומי על ידי בעל האתר, אלא על ידי תכנות בעזרת מתכנת HTML או איש מקצוע בתחום, בד"כ אתרים סטטיים ישמשו אתרים שיש בהם פחות צורך עדכון ושינוי ביומיום.

בניגוד לכך **אתר דינמי** הוא אתר שניתן לערוך אותו ולשנותו על ידי מערכת לניהול תוכן, מערכת שמקלה על ההתעסקות והעדכון השותף של תוכן האתר.

ייתכן ובונה אתרים מסוים יבקש להתמקד בעיסוקו על בניית סוג מסוים של אתרים.

בכל מקרה, ההבנה בHTML היא הכרחית לכל המעוניין לגעת בתחום הפיתוח ובניית אתרים מכל סוג שהוא, HTML הוא הדבר הראשון שילמד כל אחד שרוצה להתמקצע בתחום האתרים.

תרגיל 1 - ענו על השאלות הבאות:

◀ מהו פרוטוקול http?

◀ מהו פרוטוקול ftp?

◀ מהי כתובת ip?

◀ מהו פרוטוקול dns?

◀ מהו שמו של הפרוטוקול המתרגם את שמות התחומים המילוליים לכתובת ip?

◀ מהו שמה של התכנה המתרגמת את הקוד של האתר ומציגה ללקוח לתצוגה גרפית.

מה זה בכלל HTML?

HTML - ראשי תיבות של **Hyper Text Markup Language** - שפת סימון טקסט מקושר.

שפה המשתמשת בהיפר טקסט, בטקסט עשיר המכיל קישורים.

Html היא אינה שפת תכנות אלא שפת סימון.

היא משמשת לתצוגה ועיצוב דפי אינטרנט ותוכן לתצוגה בדפדפן.

שפת HTML היא שפת התגיות המרכזית בעולם האינטרנט. מהווה שלד לרב עמודי התוכן באינטרנט. היא מגדירה אילו אלמנטים יופיעו בעמוד האינטרנט, ובאיזו צורה יוצגו.

שפה זו מאפשר עיצוב תכון בצורה מהירה, קלה ללימוד באופן יחסי וקלה לכתיבה.

HTML תוכננה לעבוד על כל מחשב, מכל סוג היא מותרת לשימוש על ידי כל מפתח אתרים, ללא צורך ברכישת זכויות יוצרים מחברה כלשהי, וניתנת לקריאה בכל סוגי המערכות. שפת HTML מאפשר ליצור קישורים למסמכים HTML נוספים או לסוגים שונים של נתונים.

היא תומכת במולטימדיה ומאפשרת להציג נתונים בעזרת: טקסט, צליל, תמונה, וידאו ועוד.

תקן HTML

HTML משמשת תקן ברשת האינטרנט, כלליה מוגדרים על ידי ארגון תקינה בינלאומי, W3C, בניית אתר אינטרנט על פי תקני הרשת גורמת לכך שדפדפנים שונים יציגו אתרים באופן דומה. באופן זה נחסך הצורך לכתוב מספר גרסאות לכל אתר.

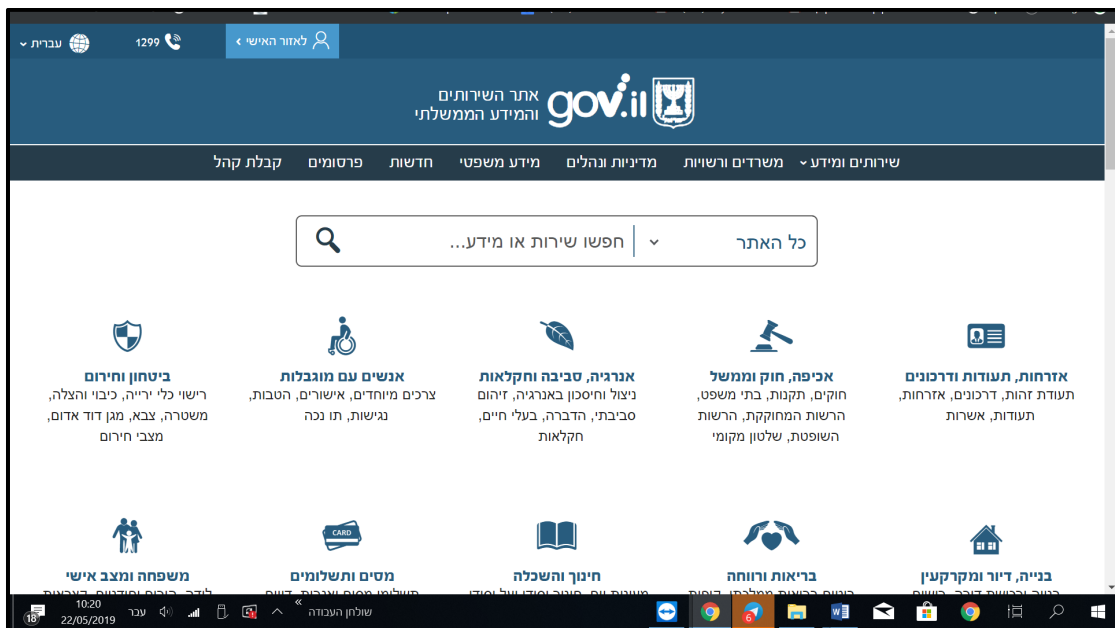
גרסת html5 היא הגרסה האחרונה שהושקה ב-28 לאוקטובר 2014.

שפת HTML אינה דורשת סביבת עבודה ייעודית על מנת לעבוד אתה. היא מורכבת מפקודות טקסט פשוטות הנכתבות במבנה של תגיות.

מאחורי הקלעים

על מנת לצפות בHTML שמאחורי הקלעים באתר חי, נפעיל את הדפדפן ונגלוש לאתר כלשהו, לדוגמה נגלוש לאתר השירותים והמידע הממשלתי:

<https://www.gov.il/>



בלחיצה ימנית באזור כלשהו בדף האינטרנט ובחירה באפשרות **הצג קוד מקור** או **צפה במקור הדף** (או באנגלית view source), נצפה בקוד ה-HTML של הדף.

```
1 <!DOCTYPE html>
2
3
4
5 <html lang="he">
6 <head>
7   <title>gov.il - האתר החדש לשירותים ולמידע ממשלתי </title>
8
9
10  <meta charset="utf-8">
11  <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=yes">
12  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
13  <meta name="google-site-verification" content="pg97KL_XTGAZ_H14HRc7DWJ5_OApykYD3nBCVFY0tce" />
14  <meta name="referrer" content="unsafe-url">
15
16
17  <meta property="og:url" content="https://www.gov.il/" />
18  <meta property="og:title" content="אתר השירותים והמידע הממשלתי" />
19  <meta property="og:description" content="אנציקלופדיית האינטרנט הממשלתי" />
20  <meta property="og:image" content="https://www.gov.il/Content/Images/Ver_2/default_logo.png" />
21
22  <meta property="og:site_name" content="GOV.IL" />
23  <meta property="og:type" content="article" />
24  <meta name="twitter:card" content="summary" />
25
26  <link rel="shortcut icon" type="image/ico" href="/Content/Images/favicon.ico" />
27  <link rel="canonical" href="https://www.gov.il/" />
28
29  <!--Add to Home Screen Popup-->
30
31
32
33
34
35
36
37  <meta name="page-title" content="gov.il דף הבית" />
38
39
40
41  <script type="text/javascript" src="/ruxitegentjs_ICA2Q5Vfgjartx_10167190509125344.js" data-dtconfig="rid=RID_2418|rpId=161070405|domain=gov.il|reportUrl=rb_c759d597-9c18-4dc6-89d5-5dfeaddc2986|app=4177d9cfa64243af|featureHash=ICA2Q5Vfgjartx|doNotDetect=kyw,kvd|msl=153600|srns=25000|rdnt=0|uxrgrce=1|bp=3|srms=1,1,...|uxrgcm=100,25,300,3;100,25,300,3|dp
```

מה אנחנו רואים בתצוגת הדף שמאחורי הקלעים?

שורות שורות שנפתחות בשני משולשים פותח וסוגר < > (גדול מ... וקטן מ...) ובתוכם מצוינת הגדרת התגית. לאחר מכן מופיע תוכן התגית, השורות הללו מסתיימות שוב בסוגריים משולשות כשבניהם הסימן / - לוכסן ושם התגית.

אם נדפדף בתצוגה שקבלנו נבחין, שכל התכנים בדף ממוקמים בין תגיות שונות.

תרגיל 2

1. גלשו לאתר כרצונכם.
2. הציגו את מאחורי הקלעים של אתר (לחיצה ימנית בעכבר < צפה במקור הדף).
3. תארו את מבנה השורות הנראות בתצוגה זו.

תגיות

מה זה תגיות, מה תפקידן, מבנה התגיות?

שפת HTML, היא שפת תגיות, שפה הבנויה מתגיות תגיות, התגיות הינן הוראות לדפדפן כיצד להציג את התוכן שבדף, לדוגמא תגית H1, הינה תגית המכריזה על תוכן שיוצג בעיצוב המוגדר לכותרות ראשיות, היא אומרת לדפדפן כאן תציג כותרת ראשית, דוגמא לכותרת ראשית:

`<h1>זו כותרת ראשית</h1>`

דוגמא נוספת, התגית a אומרת לדפדפן, הטקסט שבתוך התגית a הוא קישור לדף או אתר מסוים, התגית a אומרת לדפדפן שהטקסט שבתוכה יהפוך לקישור, ומגדירה בתוכה את ניתוב הקישור:

`זו דוגמא לקישור`

לרב התגיות יש מבנה קבוע הנראה כך:

`<סוף תגיות/> תוכן <תחילת תגית>`

תגית הבנויה מזוג תגיות

מבנה התגיות בנוי מזוג של תגיות (תגית פותחת ותגית סוגרת), שיחדיו מרכיבות את התגית הפותחת מורכבת משם התג ממקום בין סוגריים משולשים (הסימונים: גדול מ... קטן מ...) לדוגמא: `<a>`, אחר התגית הפותחת יופיע התוכן המוכל בתגית, ולסיום התגית נסגרת עם תגית נוספת הזזה לזו הפותחת בתוספת הסימון `/>` - לוכסן לדוגמא: ``.

תגים הבנויים מתגית בודדת

חלק קטן מהתגיות בHTML אינו מורכב מזוג תגיות אלא מתגית בודדת.

תגיות אלו הן תגיות שלא מכילות תוכן, אלא התגית הן התוכן בעצמו.

תגיות אלה בנויות משם התגית מוכלת בין סוגריים משולשות, בחלקן לפני הסוגר המשולש הסוגר את התגית, יופיע הסימן `/>` - לוכסן. לדוגמא: `
` (תגית המשמשת לירידת שורה).

וחלקן יופיעו רק כתגית בודדת לדוגמא התגית המשמשת לקישור לתמונה להצגה, תגי meta, ג"כ בנויים כתגית בודדת (יורחב עליהם בהמשך...).

חלוקת התגיות לסוגים שונים - סוגי תגיות

ניתן לחלק את התגיות למספר סוגים על פי תפקידן:

- **תגיות תצוגה, תגיות עיצוביות** - באופן כללי מומלץ לתת את העיצוב לכל האלמנטים באמצעות גליון CSS בלבד (יפורט בהמשך) ולא להשתמש בתגיות עיצוב, תגיות תצוגה, תגיות התצוגה הן מס' תגיות המשפיעות על אופן תצוגת התוכן, כמו התגית המגדירה הדגשה לתוכן שבתוכה, התגית <i> המגדירה שהטקסט שבתוכה יוטה.
- **תגיות מבניות** - תגיות המשמשות להוספת טקסטים לעמוד ה-html לדוגמא, התגית p תוסיף במסמך פסקת טקסט.
- תגיות המארגנות את מבנה מסמך ה-html, מה שעוזר מאד להבין את מבנה המסמך, ותורם מבחינת קידום האתר.
- **תגיות קישורים** - תגיות שתפקידן להגדיר לתוכן שבתוכן קישור למיקום אחר באתר, או למיקום שונה ברשת.
- **תגיות הוספה** - תגיות המשמשות להוספת אלמנטים שונים לעמוד, כגון הוספת תמונות, קבצי שמע וקבצי מדיה שונים.
- **תגיות טפסים** - תגיות שתפקידן להרכיב טופס לקבלת נתונים מהמשתמש, תגיות טופס כוללות תגיות להזנת מידע, לסימון מידע, לבחירת ערך מתוך רשימה ועוד.
- **תגיות סמנטיות** - תגיות התוחמות אזורים בדף, המאגדים מספר אלמנטים השייכים לאותו אזור, כגון התגית body, התגית head, והתגית div, ב-html5 קיימות מספר תגיות חדשות לתחימת אזורים כמו התגית article והתגית footer, התגית nav ועוד (יפורט בהמשך),

מבנה מסמך html בסיסי ותכונות של תגים

מבנה בסיסי של מסמך HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>עמוד עמוד</title>
```

```
  </head>
```

```
  <body>
```

זה גוף המסמך, כאן יבוא כל הטקסט והתוכן שנרצה להציג במסמך לדוגמא:

```
    <h1>זו כותרת לתפארת לתוכן העמוד</h1>
```

```
    <p>זו פסקה שנכתוב בה דברים מעניינים</p>
```

```
  </body>
```

```
</html>
```

השורה הראשונה **התגית DOCTYPE** (בתרגום לעברית סוג מסמך) חייבת לפתח כל מסמך HTML, תפקידה להודיע לדפדפן באיזו גסת HTML כתוב המסמך שהוא עומד לקרוא, **<!DOCTYPE html>** זו צורת ההגדרה ל - HTML5 חשוב לשמור על אותיות גדולות וקטנות.

השורה השנייה התג **<html>** מורה לדפדפן לפתוח מסמך מסוג HTML, תג זה הינו התג הראשי במסמך, ובין התגית הפותחת שלו לתגית הסוגרת שלו ימוקמו כל תגי הHTML האחרים.

השורה השלישית התג **<head>** משמש אותנו בכדי להגדיר את אזור ראש המסמך (head - ראש), תגית head תעטוף רק תגים שמתארים את המסמך וקובעים את תצורתו, מלבד התגית **<title>**, שאר התגים שימוקמו בה הם אינם אלמנטים במסמך אלא לרב הם תגים המכילים מידע שאינו מוצג במסך, אך משמשים להגדרת המסמך על ידי מנועי החיפוש. (כמו תגיות Meta וכן תגית script לקישור לסקריפט של javascript לדוגמא, או תגית style להגדרת סגנונות עיצוב לאלמנטים בעמוד או הגדרה לדפדפן היכן למצא את קבצי הcss שלנו).

התג **<title>** בתוכו נגדיר את כותרת המסמך, כותרת המסמך תוצג בראש הדפדפן (בלשונית הדף), תגית זו תשב באופן קבוע בתוך תגית head .

תגי **<meta>** הם תגים מיוחדים. הם לא מציגים דבר בדף, אלא משמשים להעברת מידע על הדף לדפדפן של הגולש, למנועי חיפוש, וכיוצא בזה (יורחב בהמשך...)

המאפיין charset - קידוד, מגדיר את אופן קידוד התווים במסמך, קידוד התווים שקבענו בתגית meta הוא קידוד תווים עבור מסמך HTML, UTF-8 - קידוד Unicode, קידוד המגדיר את המסמך שלנו לעבוד עם טקסטים לועזיים וטקסטים בעברית בו זמנית, חוסר הגדרה זו עלולה להציג לעיתים את האתר שלנו בג'יבריש - טקסט תקול, במקום להציג את התוכן הרצוי.

התג **<body>** הוא גוף המסמך, (body - גוף), בתוכו ימוקמו כל התגים האחרים המוצגים בדף - כל תוכנו של הדף.

התגיות **<h1>**, **<p>** הן תגיות המציגות תוכן, נרחיב עליהם בהמשך....

כל תגית כזו שנפתחת ונסגרת ומסוגלת להכיל בתוכה נתונים מכונה גם אלמנט.

שימו לב: לעקרון ההזחה, לצורת ההירארכיה של התגיות, התגיות הראשיות מוצמדות לצד הימני של מסמך הHTML, ותגיות המוכלות בתוך תגיות אחרות, מוזחות - נדחפות מעט פנימה, מבנה זה חשוב על מנת לשמור על מבנה בהיר ומובן.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>כותרת עמוד</title>
  </head>
  <body>
    זה גוף המסמך, כאן יבוא כל הטקסט והתוכן שנרצה להציג במסמך לדוגמא:
    <h1>זו כותרת לתפארת לתוכן העמוד</h1>
    <p>זו פסקה שוכחור רה דריח מעוייח</p>
  </body>
</html>
```

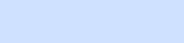
חשוב לדעת: כל תו ותו בתגיות הינו חשוב, בהשפעת המבנה של התגיות הדפדפן מבין את מבנה ותוכן העמוד. החסרת או הוספת אפילו תו אחד בטעות בתגית, הדפדפן לא יבין מה עליו לבצע והוא לא יציג את התוכן באופן הנדרש.

בכתיבת HTML יש לכתוב את שמות התגיות ומאפייניהם באותיות קטנות (למעט תגית ה-DOCTYPE).

תחילת העבודה - סביבת העבודה

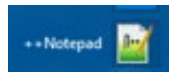
לכתיבת דפי html לא נשתמש במעבד תמלילים אלא נשתמש בעורך טקסט כמו atom, notepad++ וכדו', ההבדל בין עורך טקסט למעבד תמלילים הוא בכך שהעורך טקסט מקבל רק טקסטים, אין אפשרות של הוספות תמונות או כל מדיה על גבי העורך, אין אפשרויות עיצוביות לטקסט בעורך הטקסט, מה שקיים במעבד תמלילים....

להתקנת עורך הטקסט notpa++, גשו לקישור הבא: <https://notepad-plus.org/download>

בחרו בלחצן DOWNLOAD  להורדת קובץ ההתקנה.

לחצו לחיצה כפולה על הקובץ שהורדתם .

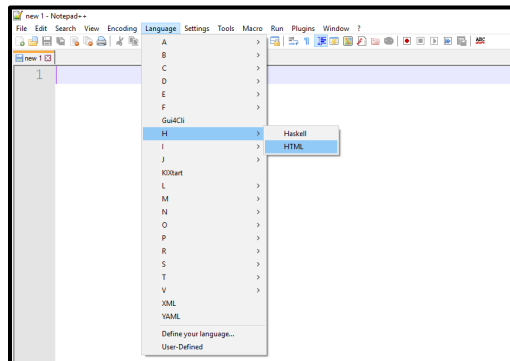
התקינו את ה-notepad++ במחשבכם.



1. פתחו את תכנת ה notepad++ שהתקנו\

2. נגדיר שאנו עומדים לכתובת תכנית בHTML כך:

בתפריט ניגש ל Language (שפה) < האות H < HTML.



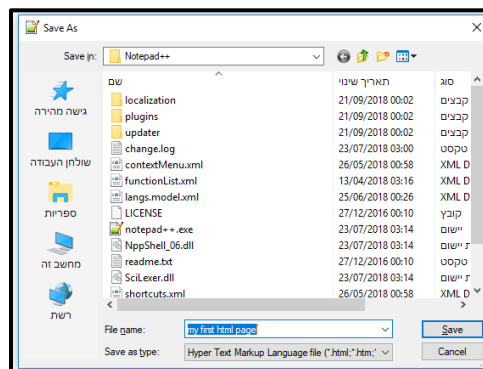
1. נשמור את הקובץ בתיקיית העבודות שלנו:

2. בתפריט נבחר `file> sava as...`

3. ננתב למיקום התיקייה שלנו, ששם נשמור את הקובץ.

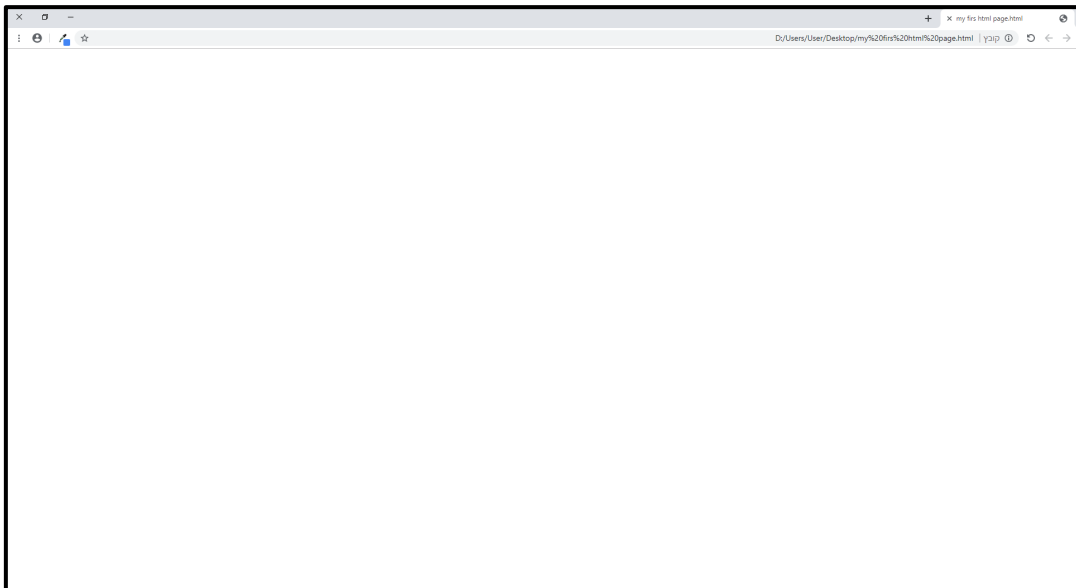
4. בתיבה file name, נקליד את שם הקובץ. (שם הקובץ צריך להיות באנגלית).

5. וכמובן נלחץ על save, לשמירת הקובץ.



לאחר השמירה, נקבל דף אינטרנט ריק, הנה הקובץ ששמרנו,

כשנפתח אותו בקליק כפול, נצפה בעמוד ריק, חסר תוכן, כיוון שעדין לא הכנסנו בו כל תוכן שהוא.



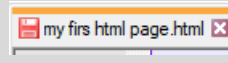
נחזור לnotepad נכניס תוכן לדף שלנו:

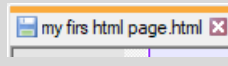
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>העמוד שלנו</title>
  </head>
  <body>
    <p>שלום עולם</p>
  </body>
</html>
```

לאחר שהקלדנו תוכן לאתר בעורך הטקט, נשמור את השינויים (file>save).

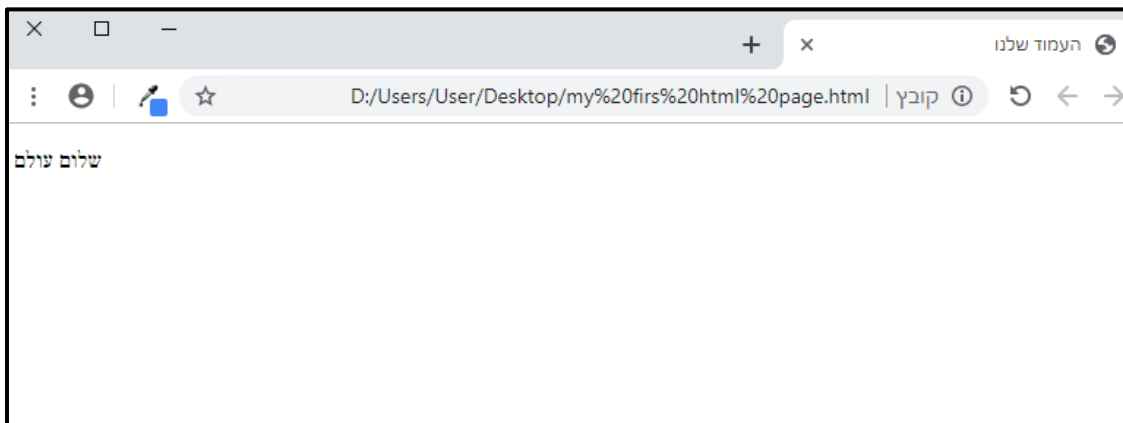
שימו לב:

קובץ שבצענו בו שינויים ועדיין לא שמרנו אותם, הדיסקט המאוויר בלשונית שם הקובץ

בסמוך לשם הקובץ יופיע בצבע אדום כך: 

לאחר שנשמור את השינויים הדיסקט ישנה את צבעו לכחול כך: 

נפעיל שוב את הקובץ ששמרנו, כעת נצפה בדף בעל תוכן, נשים לב שגם לשונית שם הדף קבלה את השם שהגדרנו ב<title>.



פתיחת מסמך שמור לעריכה

על מנת לערוך שוב קובץ שסגרנו בעורך, לשנות דברים בעמוד או להוסיף עדכונים:

1. ניגש לקובץ ששמרנו.

2. נלחץ לחיצה ימנית בעכבר.

3. נבחר באפשרות: Edit with Notepad ++.

נעבור לתרגול, נבנה דף עם תוכן ונצפה בדפדפן בתוכן שהכנסנו.

1. פתחו את תכנת ה- notepad++
2. הגדירו את הקובץ כקובץ HTML.
3. שמרו את הקובץ בתיקייתכם האישית.
4. הקלידו את התוכן הבא במסמך:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>עמוד ראשי</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>אתר טוב</h1>
```

```
    <p>
```

כדי לבנות אתר טוב ומעניין, דבר ראשון צריך לתכנן, מהו התוכן שיוצג בכל עמוד, מה חשוב יותר ומה חשוב פחות, את הדברים החשובים יותר, נדגיש מבחינה עיצובית ונמקם אותם במקומות בולטים יותר, נדגיש את העיקר, את הטפל נמקם במקומות פחות חשובים בעמוד.

```
    </p>
```

```
    <p>
```

כדי לבנות אתר טוב ומפיק תועלת, נתכנן את חלוקת התוכן לתגיות, נמקם כל תוכן בתגית המתאימה לו, תוכן קישורי נמקם בתגית קישור, את הכותרות נכניס לתגיות כותרת לפי סדר חשיבותן ומעלתן.

```
    </p>
```

```
    <p>
```

אתר טוב, הוא אתר שמשיג את מטרתו, כדי להניעה את הגולשים לפעולה עלינו לחשוב מראש מהי מטרת האתר ולחשוב מהי הדרך הטובה לגרום לגולשים לפעול על מנת שנשיג מטרה זו.

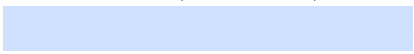
```
    </p>
```

```
    <p>
```

דף טוב ומעניין, הוא דף שמכיל לפחות שלשה כותרות, לפחות שני קישורים ולפחות תמונת מדיה אחת.

```
    </p>
```

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

5. שמרו את השינויים.
6. פתחו שוב את הדף לצפייה בדפדפן.
7. הטקט בעמוד מיושר משמאל לימין, על מנת שהתוכן יוצג מימין לשמאל, הוסיפו את ההגדרה: `dir="rtl"` לתגית `body` כך:

8. שמרו את השינוי, וצפו שוב במסמך שעודכן.

למדנו להכיר מה זה מסמך `html` בסיסי, למדנו להכיר את השלד שלו.

כעת נלמד איך להרכיב על גבי השלד מעט אבני בנין, כך שהדף שלנו יתחיל להתמלא ולקבל צורה ותוכן.

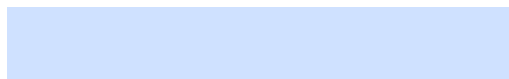
הערות בדף HTML

לפעמים נרצה להוסיף בדף `html` טקסט שלא יוצג בדף לגולש, אלא טקסט שהוא הערה שנרצה לציין לעצמנו על גבי מסמך ה-HTML.

לשם כך נשתמש נוסף את התוכן בצורה כזו שיוגדר כהערה, תוכן שמוגדר כהערה, אינו מוצג על גבי הדף לגולש.

להגדרת תוכן שיוגדר כהערה נוסף בתחילת השורה את הסימן ->!
בהערה ונסיים את בסגירה עם קו אמצע, קו אמצע וסוגר משולש -->

ההערה תראה כך:



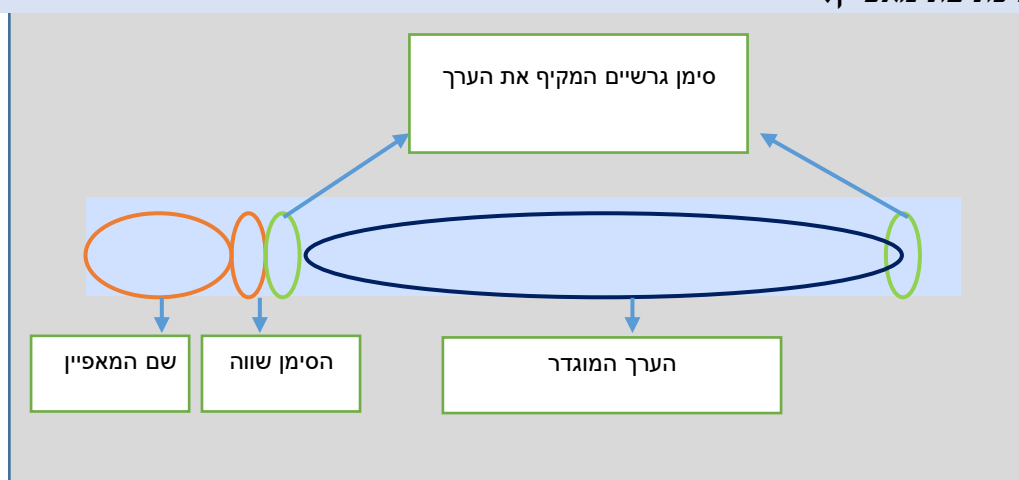
תרגיל 4

1. פתחו את מסמך `html` שיצרת בתרגיל 1
2. הוסיפו בשלשה מקומות במסמך הערות כרצונך.
3. שמרו את המסמך.

Attributes - מאפיינים, תכונות של תגים

למרבית התגיות יש מאפיינים נוספים המוגדרים בתגית, תפקידם של המאפיינים להורות לדפדפן הוראות שונות, ולספק מידע נוסף על האלמנט, התכונות מופיעות תמיד בתג הפותח, בתג ההתחלתי. המאפיינים באים בזוגות של שם וערך המונח בתוך גרשיים, כשבניהם מופיע הסימן שווה "=".

צורת כתיבת מאפיין:



כמה דוגמאות למאפיינים

נציג כמה תגיות הנושאות מאפיינים שונים, בהמשך נלמד בהרחבה על התגיות הללו

← בתגית קישור יופיע המאפיין href, מאפיין זה מציין לדפדפן את יעד הקישור, להיכן הקישור יפנה.

לאתר השירותים והמידע הממשלתי

← בתגית תמונה יופיעו כמה מאפיינים, המאפיין src, המציין את שם התמונה ונתיב התמונה במידת הצורך), המאפיינים width ו-height המציינים את רוחב וגובה התמונה והמאפיין alt (קיצור של המילה alternative - אלטרנטיבה, חלופה), יציין טקסט חילופי במידה והדפדפן יתקל בקשיים להצגת התמונה ולציון מהות התמונה לקוראי מסכים, בדרך זו בעלי לקות של כבדות ראייה יכולים לשמוע את תוכן האלמנט, שאינם מסוגלים לצפות בו.

```

```

← בתגית <html> ניתן להגדיר את שפת המסמך, באמצעות המאפיין lang (קיצור של המילה language - שפה), הגדרה זו חשובה ליישומי נגישות (קוראי מסך) ומנועי חיפוש.

```
<html lang="he">
```

ניתן לצפות ברשימה המלאה של הגדרת השפות השונות:

https://www.w3schools.com/tags/ref_language_codes.asp

← ניתן להוסיף לאלמנטים השונים במסמך html מאפיין המוסיף כעין כותרת צפה המכילה הסבר קצר ומוצגת בעת מעבר על האלמנט.

```
<p title="אני הסבר קצר">
```

זאת פסקה

```
</p>
```

שים לב: ערכי התכונות ימוקמו בד"כ בתוך זוג מירכאות כפולות, אך אם התכונה מכילה מירכאות כפולות יש להכניס את הערך לתוך גרשיים ולא מירכאות כפולות.
לדוגמא:

```
<a title="אם אתה 'לא מסכים' לחצן כאן">
```

עוד ועוד מגוון של מאפיינים, שעם חלקם נפגש במהלך הקורס...

תרגיל 5

← רשמו מהו המאפיין המציין לקורא המסכים באיזו שפה כתוב תוכן דף ה-html ?

← רשמו מהו המאפיין שנציין בו טקסט חילופי לתמונה?

◀ רשמו מהו המאפיין שנשתמש בו על מנת להציג טקסט קצר על אלמנט בדף?

◀ רשמו מהו המאפיין שנשתמש בו לציון יעד הקישור?

תגיות meta ותגיות נפוצות (כותרות ופסקאות)

תגיות meta - הרחבה

תגי מטה הם תגים מיוחדים שאינם מציגים דבר לגולש, אלא תפקידם לתת מידע על האתר לדפדפן ולמנועי החיפוש, רב תגיות המטה יקבלו שני מאפיינים עיקריים: name ו- content, כשהמאפיין name, יכיל את מתאר תגית המטה, אם זו תגית מטה לתיאור הדף, לסיכום מילות מפתח, להצגת שם המחבר וכדו', המאפיין content, יקבל את התוכן לתיאור.

מספר דוגמאות:

המאפיין Description

`<meta name="description" content="רפואה על שיטות מסמך הסבר על שיטות רפואה" >`

(Description - תיאור) מאפיין זה משמש לתיאור מילולי קצר על התוכן של הדף, התיאור אינו אוסף של מילות מפתח אלא תוכן שנקרא ברצף, וחשוב להקפיד שהתוכן יהיה קשור וישקף את תוכן הדף. אורך הטקסט המומלץ להכניס בתוך התגית הזו, הוא עד 20 מילים. ובחלק ממנועי החיפוש, הוא התוכן שיופיע בדף התוצאות.

המאפיין keyword

`<meta name="keywords" content="HTML,CSS,XML,JavaScript">`

(keyword - מילות מפתח) מאפיין זה משמש להגדרת אוסף מילות מפתח המופרדות בפסיקים ביניהם, חלק ממנועי החיפוש מתייחסים לתגית הזו ומתבססים עליה בעת החיפוש, לתת מענה לחיפושי הגולש לפי מילות אלה, מומלץ לשלב מילות מפתח המופיעות באתר, ובנוסף מילות מפתח קרובות או מילות מפתח הכתובות בצורה שגויה, על מנת לתת מענה לחיפושים קרובים של המשתמש, כגון שגיאות כתיב וכדו'.

המאפיין author

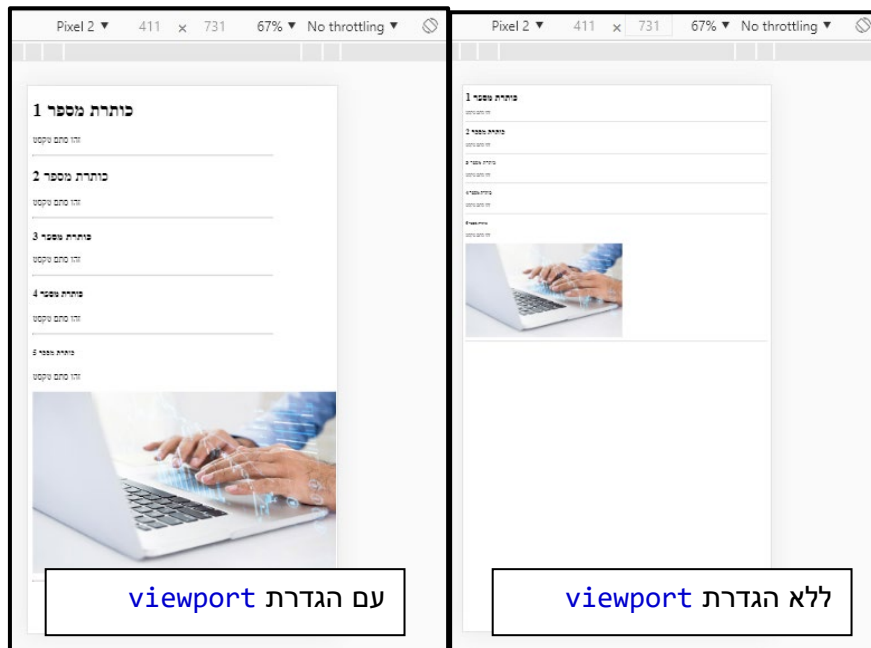
```
<meta name="author" content="ג'ון סמיט">
```

(author - מחבר) תג זה מציין את שמו של מחבר הדף.

המאפיין viewport

```
<meta name="viewport" content="width=device-width,initial-cale=1.0">
```

מאפיין זה מספק הוראות לדפדפן כיצד לשלוט במידות הדף ובקנה המידה של האלמנטים שבדף, החלק הראשון המוגדר בתוכן, החלק `width=device-width`, הוא החלק שקובע את רוחב הדף לרוחב המכשיר, החלק השני קובע את רמת הזום הראשוני כשהדף נטען לראשונה על ידי הדפדפן, מאפיין זה עוזר מאד בהתאמת הדף לסוגי המכשירים השונים, במילים אחרות משפיע על הרספונסיביות הבסיסית של האתר.



נוכל בתמונה הזו לצפות בהבדל בין אותו הדף עם תגית המטה `viewport` / ללא.

(קובץ מצורף)

רענון הדף באמצעות תג meta

ניתן לקבוע לדפדפן לרענן את הדף לאחר פרק זמן מסוים, או לטעון דף אחר תוך זמן מסוים.

```
<meta http-equiv="refresh" content="3;url=page2.html">
```

המאפיין `http-equiv="refresh"`, מגדיר את הרענון, במאפיין `content` נגדיר את מספר השניות שלאחריו ירוענן הדף, בנוסף ניתן להגדיר כתובת `url`, נתיב לדף אחר באתר במידה ונרצה שלאחר הרענון יופיע עמוד שונה, כפי שאנו רואים בדוגמא שהדף שיוצג לאחר הרענון הוא דף מספר 2.

דוגמא נוספת:

```
<meta http-equiv="refresh" content="5">
```

שורה זו מגדירה רענון לדף כל 5 שניות, בלי לעבור לדף שונה.

על תגית המטה `<meta charset="utf-8">`, פרטנו במבנה הבסיס של דף `html`.

תרגיל 1

1. פתחו את המסמך `tar-1.html`.
2. הוסיפו למסמך משפט תיאור בתגית `meta` מתאימה.
3. הוסיפו למסמך מילות מפתח בתגית `meta` מתאימה.
4. הוסיפו למסמך את שמכם כשם מחבר, השתמשו בתגית `meta` מתאימה.
5. שמרו את המסמך.

תרגיל 2

1. פתחו את תכנת ה-`notepad++`.
2. הגדירו את הקובץ כקובץ `HTML`.
3. שמרו את הקובץ בתיקייתכם האישית.
4. הקלידו את התוכן הבא במסמך:

```
<!DOCTYPE html>  
<html lang="he">  
  <head>
```

```
<meta charset="utf-8">
<title>אגן החיות - עמוד ראשי</title>
</head>
<body>
  <h1>ברוכים הבאים לאתר האינטרנט הראשון שלנו</h1>
  <p>
    מיד תועברו לתוכן האתר.....
  </p>
</body>
</html>
```

5. שמרו את המסמך בשם: page1.html

6. צרו מסמך html נוסף

7. הקלידו בו את התוכן הבא:

```
<!DOCTYPE html>
<html lang="he">
  <head>
    <meta charset="utf-8">
    <title>אגן החיות</title>
  </head>
  <body>
    <h1>ברוכים הבאים לגן החיות</h1>
    <h2>האריה</h2>
    <p>
    </p>
    <h2>הנמר</h2>
    <p>
    </p>
    <h2>הדב</h2>
    <p>
    </p>
  </body>
</html>
```

8. חפשו בויקיפדיה את ההגדרה המתאימה לכל בעל חיים: האריה, הנמר והדב.
9. העתיקו מהויקיפדיה לכל בעל חיים את הגדרתו והדביקו בתוך התגית `<p>` הממוקמת תחת שמו.
10. שמרו את המסמך בשם `page2.html`.
11. הגדירו שעמוד `Page1` יטען את עמוד `page2`, לאחר 4 שניות מטעינתו. (היעזרו כמובן בתגית ה-`meta` לרענון הדף).
12. שמרו את השינוי ב-`page1`.
13. הפעילו את הקובץ `page1` בדפדפן, בדקו שהקובץ `page2` נטען כראוי.

נרחיב על תגיות נפוצות במסמך html

כיוון שחוץ מקהל הגולשים הצופה באתר שלנו, אתרנו גם חשוף לעיניים של "גוגל" ושאר מנועי החיפוש, יש לכך משמעות מרובה, ומחייבת התייחסות, על מנת שאתרנו יקודם ויפיע במקום כמה שיותר גבוה ברשימת תוצאות מנועי החיפוש, ועל מנת שהדף יראה "טוב" בעיני גוגל ודומיו מומלץ לשתול בתוכו לפחות שלוש רמות של כותרות, טקסט רץ של פסקאות, ותגיות של קישורים, אלמנטי הכותרות שאנו שותלים בדף צריכות להתייחס ב"כובד ראש", ולשמש כ"חלון ראווה" המלמד את מנועי החיפוש מהו תוכן האמיתי של הדף. תחילה נרחיב על תגיות הכותרות:

<h1>-<h6> - תגיות כותרת

קיימות שש רמות של כותרות, כשהרמה הראשונה מיועדת לכותרת המרכזית ביותר בעמוד, כותרת ברמה ראשונה תופיע פעם בודדת בעמוד ולא יותר מכך, כותרות נוספות יופיעו ככותרות משניות, ברמות נוספות על פי סדרן, כלומר לא נשלב כותרת ברמה שלישית <h3>, תחת כותרת ברמה ראשונה <h1>, אלא כותרת ברמה שלישית תופיע תחת כותרת מהרמה הקודמת (רמה שניה <h2>), את הכותרות בעמוד נתחום בתגיות <h1>-<h6>, (h - קיצור המילה heading - כותרת) כשהתגית <h1>, משמשת לכותרת העיקרית ביותר במסמך ושאר הכותרות לפי מעלת הכותרת, מבחינה ויזואלית הכותרות יוצגו בצורה בולטת כשהתגית h1 הוא הבולט ביותר והגדול ביותר, ושאר התגיות בולטות מעט פחות לפי סדרן, יש לא מעט תחליפים לתצוגה ויזואלית של הנתונים שיופיעו ככותרות, אך מבחינה סמנטית אין להן תחליף והן חייבות ללמד על תוכנו האמיתי של הדף.

נציג את הקוד הבא, כך תראנה הכותרות ברמותיהן השונות:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>כותרת עמוד</title>
  </head>
  <body>
    <h1>כותרת מספר 1</h1>
    <h2>כותרת מספר 2</h2>
    <h3>כותרת מספר 3</h3>
    <h4>כותרת מספר 4</h4>
    <h5>כותרת מספר 5</h5>
  </body>
</html>
```

תצוגת הקוד בדפדפן

1 כותרת מספר

2 כותרת מספר

3 כותרת מספר

4 כותרת מספר

5 כותרת מספר

התגית <hgroup> - קבוצת כותרות

תגית זו, היא תגית המאגדת מספר כותרות, כמו לדוגמא במבנה הבא:

```
<hgroup>
  <h2>כותרת ראשית</h2>
  <h3>כותרת משנית</h3>
</hgroup>
```

התגית <p> - תגית פסקה

התגית, p, היא תגית שנכניס בתוכה תוכן של פסקה (paragraph - פסקה) כשנרצה להתחיל פסקה חדשה, נסגור את תגית ה-p, ונפתח תגית p חדשה, כל פסקה תמוקם בתוך אלמנט פסקה משל עצמה.

ישנה אופציה לחתוך באמצע שורה באמצעות התגית `br` (brake row) לא משתמשים בחיתוך של שורה במקום התחלה של פסקה חדשה.

</p> פסקה המכילה מלל ותיאור אודות דברים מעניינים ומגוונים<p>

</p> פסקה נוספת לתיאור דברים נוספים ומלל נוסף <p>

התגית `<hr>`

התגית `hr`, משמשת להפרדת תוכן בדף, לחלוקת הדף בחלוקה אופקית באמצעות איור של כעין צורת קו המחלק את הדף בצורה אופקית, נשתמש בו כאשר נרצה לחלק, להפריד בין שני אזורים, תגית זו היא תגית שלא מקבלת תוכן לתוכה ומורכבת מתגית בודדת כך: `<hr>`.

לדוגמא הקוד הבא:

```
<!DOCTYPE html>
```

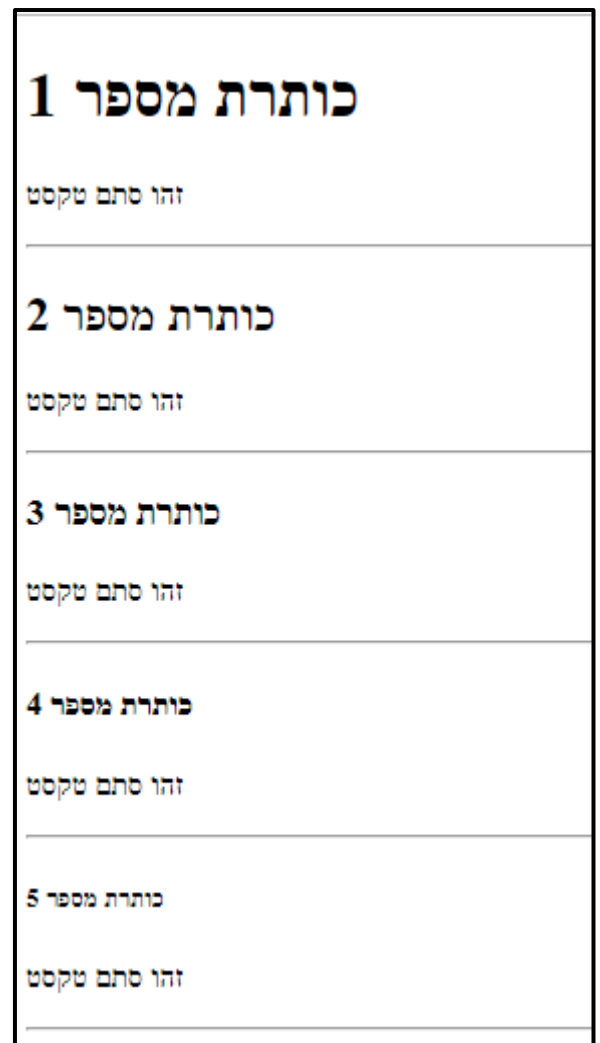
```
<html>
```

תצוגת הקוד בדפדפן


```

<head>
  <meta charset="utf-8">
  <title>כותרת עמוד</title>
</head>
<body>
  <h1>כותרת מספר 1</h1>
  <p>זהו סתם טקסט</p>
  <hr>
  <h2>כותרת מספר 2</h2>
  <p>זהו סתם טקסט</p>
  <hr>
  <h3>כותרת מספר 3</h3>
  <p>זהו סתם טקסט</p>
  <hr>
  <h4>כותרת מספר 4</h4>
  <p>זהו סתם טקסט</p>
  <hr>
  <h5>כותרת מספר 5</h5>
  <p>זהו סתם טקסט</p>
  <hr>
</body>
</html>

```



(אחרי שלמדנו על תגיות המטה תגית פסקה ותגיות הכותרת, נעבור ללמוד על תגית נוספת),

התגית `
` - תגית שבירת שורה

התגית `
` (brake row) מבצעת הורדת שורה, שבירת שורה - התגית גורמת לירידת שורה מבלי להתחיל פסקה חדשה.

לדוגמא:

תצוגת הקוד בדפדפן



`</p>
 הלך לגן
 יונתן הקטן <p>`

תרגיל 3

1. פתחו את תכנת ה-`notepad++`
2. הגדירו את הקובץ כקובץ HTML.
3. שמרו את הקובץ בתיקייתכם האישית.
4. הקלידו במסמך תוכן המכיל:
 - a. 3 כותרות, לפחות משתי רמות.
 - b. לפחות 8 פסקאות.
 - c. השתמשו בתגית `br` במידת הצורך.
5. עדכנו את שמירת הנתונים.
6. צפו במסמך שיצרתם.

תרגיל 4

1. חפשו בויקיפדיה את הערך מכוניות (דף ה-`html` מהויקיפדיה קיים גם בתיקיית השיעור).
2. העתיקו את תוכן המאמר, הכניסו כל חלק לתגית מתאימה, את הכותרות לתגיות המתאימות בהתאם להיררכיה שלהן, את הפסקאות השונות בתוך התגית `p`.
3. שמרו את המסמך.
4. צפו בתוצאה.

המשך תגיות נפוצות (קישורים)

התגית <a> - תגית קישור

דפי html הם דפים מקושרים, כך שמילים רבות ואלמנטים רבים בדף מובילים לדף אחר, או למיקום שונה במסמך, גם התפריטים באתרים, בנויים מרשימת קישורים.

התג <a> משמש להגדרת קישור (link) במסמך html, הכתובת שאליה מפנה הקישור תכתב במאפיין href, לדוגמא כך:

לאתר השירותים והמידע הממשלתי

המאפיין href בדוגמא מגדיר שהקישור יוביל לכתובת: <https://www.gov.il>, בין תגית ה-a הפותחת לתגית הסוגרת, ימוקם האלמנט שיהווה כלחצן, מקובל למקם טקסטים או תמונות.

מהו url?

URL, ראשי תיבות של המילים **Uniform Resource Locator**, url מציין את מיקום הקובץ, הדף בשרת, לכל דבר הנמצא ברשת האינטרנט קיים url משלו.

כתובת ה-url מציינת לדפדפן את הנתוב שעליו לעשות על מנת להגיע לקובץ או לדף, כתובת url מורכבת בד"כ מפרוטוקול התקשורת + כתובת האתר, כתובת השרת + כתובת הדף או הקובץ בתוך האתר.

לדוגמא:

https://www.gov.il/he/departments/ministry_of_interior

בדוגמא זו כתובת ה-url מורכבת מהנתונים הבאים:

פרוטוקול התקשורת: https + כתובת האתר: www.gov.il + ניווט למיקום דף פנימי בתוך אתר השירותים הממשלתיים: .he/departments/ministry_of_interior

תרגיל 6


1. פתחו מסמך html חדש.
2. הוסיפו קישור ל-3 אתרים שונים.
3. ציין את שם האתר בתיאור מילולי בין תגית ה-a הפותחת לתגית ה-a הסוגרת.

קישור מדף אחד לדף אחר באתר

על מנת ליצור קישור בין דפי האתר יש צורך להגדיר במאפיין href את הקישור לדף, הקישור יתואר על ידי שם הדף וסיומ הדף. לדוגמא: אם נרצה לקשר לדף בשם about.html הקישור יראה כך:

```
<a href="about.html">אודותינו</a>
```

תרגיל 7

1. צרו אתר לכלי זמר.
2. האתר יכלול 4 דפים:
 - (1) דף הבית - יכיל הודעת פתיחה: ברוכים הבאים לאתר הכלי זמר.
 - (2) 3 דפים המכילים תוכן על כלי נגינה שונים, ניתן לחפש תוכן על כלי הנגינה שתבחרו בויקיפדיה או בכל אתר אחר. (מצורפת תיקייה עם תוכן מתוך הויקיפדיה, כמו"כ ניתן לחפש בויקיפדיה).
 3. הניחו בראש כל העמודים קישורים לכל עמודי האתר, בתצוגה זה יראה כך:

 4. שמרו את תוכן המסמך, וצפו במסמך שהכל עובד כראוי.

קישור פנימי בתוך הדף

במקרים רבים נרצה ליצור קשר ממקום אחד בדף למקום אחר, כמו לדוגמא מסמך ארוך המכיל בתחילתו תוכן עניינים ונרצה שהתוכן עניינים יהיה דינאמי ויוביל למיקום המדויק בעמוד, או באתרים רבים הבנויים מעמוד יחיד (אתרי one-page), התפריט העליון יוביל לחלקיו השונים של הדף.

כדי ליצור קישור ממקום אחד בדף למקום אחר עלינו לבצע שתי פעולות:

1. **הגדרת מזהה לאלמנט** אליו נרצה לקשר, לדוגמא כותרת בעמוד שנרצה לקשר אליה נוסף לה מאפיין בשם id כשהערך שלו יהיה ערך שנבחר, טקסט באנגלית כלשהו, עדיף לתת לאלמנט id המביע את משמעותו של האלמנט ולא סתם טקסט ללא משמעות.

2. **הגדרת הקישור** תוך שימוש במזהה שהוגדר בתוספת הסימן סולמית.

לדוגמא לפנינו קוד הhtml הבא:

```
<!DOCTYPE html>
<html>
  <head>
    <title>קישורים פנימיים</title>
  </head>
  <body dir="rtl">
    <h1>מאמר ארוך ומעניין</h1>
    <p>המאמר מחולק למספר חלקים</p>
    <h2>מבוא</h2>
    <p>פרק מבוא המביא רקע על הנושא שיפורט במהלך המאמר</p>
    <h2>פרק 1</h2>
    <p>פרק ראשון במאמר בו יפורט ויבואר תוכן מעניין ביותר</p>
    <h2>פרק 2</h2>
    <p>פרק נוסף המרחיב ומוסיף עוד חלקים ורבדים למאמר</p>
    <h2>סיכום</h2>
```

```
</p>פרק סיכום בו יסוכמו מסכנות חשובות העולות מתוך חלקי המאמר</p>
</body>
</html>
אנחנו רוצים להגדיר בתחילת המאמר, קישורים המובילים לחלקיו השונים, לשם כך
נגדיר id (מזהה ייחודי) לכל כותרת שנרצה לקשר אליה כך:
<!DOCTYPE html>
<html>
<head>
<title>קישורים פנימיים</title>
</head>
<body dir="rtl">
<h1>מאמר ארוך ומעניין</h1>
<p>המאמר מחולק למספר חלקים</p>
<h2 id="intro">מבוא</h2>
<p>פרק מבוא המביא רקע על הנושא שיפורט במהלך המאמר</p>
<h2 id="pre1">פרק 1</h2>
<p>פרק ראשון במאמר בו יפורט ויבואר תוכן מעניין ביותר</p>
<h2 id="pre2">פרק 2</h2>
<p>פרק נוסף המרחיב ומוסיף עוד חלקים ורבדים למאמר</p>
<h2 id="end">סיכום</h2>
<p>פרק סיכום בו יסוכמו מסכנות חשובות העולות מתוך חלקי המאמר</p>
</body>
</html>
```

לאחר שהגדרנו את המזהים, נפנה למיקומים הפנימיים שהגדרנו, ע"י התו סולמית-# בצירוף ה-id שהגדרנו לאלמנט שאליו נרצה להגיע בלחיצה כך:

```
<a href="#intro">מבוא</a><br />  
<a href="#pre1">1 פרק</a><br />  
<a href="#pre2">2 פרק</a><br />  
<a href="#end">סיכום</a><br />
```

למדריך: מצורף קובץ להצגה, קובץ בשם: links.html.

קישור מדף אחד לאזור פנימי בדף אחר

במקרים רבים נרצה לקשר קישור בדף אחד לאזור ספציפי פנימי בדף אחר.

על מנת ליצור קישור כזה עלינו לבצע שתי פעולות:

1. נגדיר תחילה באלמנט שאליו נרצה לקשר, מזהה ייחודי (מאפיין id כשהערך שלו לאחר הסימן = הוא מילה כלשהי באותיות באנגלית, נוכל לתת למזהה כל שם שנרצה) על מנת שנוכל לקשר אליו.
2. נשתמש בקישור דומה לקישור מיקומים בדף כפי שראינו קודם, רק נוסיף תחילה את כתובת הדף בו ממוקם האלמנט הספציפי שאליו אנו מעוניינים לקשר ולאחר מכן נוסיף את הסימן # ושם המזהה.

לדוגמא:

```
<a href="otherPage.html#contact">לדרכי הגעה</a>
```

שימו לב: קישור המאפשר לקפוץ למקומות שונים בדף, קישור מוביל לאזור פנימי בדף נקרא גם בשם "קישור עוגן".

תרגיל 8

1. צרו מסמך html חדש
2. צרו כותרת ראשית כתוב בה "הפרחים"
3. צרו כותרת משנית כתוב בה "פרח הגדרה"
4. צרו מספר פסקאות, כתוב בהם הגדרה למושג "פרח".

5. צרו תוכן עניינים המכיל את הערכים הבאים, בכל שורה התוכן ימוקם בתוך תגית <a> הממוקמת בתוך התגית <p> (על מנת שערך בתוכן עניינים יופיע בשורה חדשה):

a. מבנה הפרח

b. חלקי הפרח

c. צורת הפרח

d. צבע הפרח

e. האבקה

6. צרו כותרת נוספת, תוכן הכותרת: "פרחים שונים":

7. תחת הכותרת פרחים שונים יופיע תוכן עניינים נוסף:

a. סביון

b. חמציץ

c. נץ חלב

d. סאקורה

פרטו על כל פרח המתואר בכותרת לפי סדרן של הכותרות בתכני עניינים.

קשרו כל כותרת למיקום הפירוט שלה בדף.

(את ההגדרות והתכנים חפשו באינטרנט באתר ויקיפדיה, או בתכנים בתיקייה מצורפת)

תרגיל 9

1. צור מסך html חדש.

2. הוסיפו במסמך כותרת: "ברוכים הבאים לגן החיות הווירטואלי".

3. צרו 3 עמודים נוספים: בעלי חיים טורפים, עופות, בעלי חיים ימיים, שמרו את העמודים.

4. בכל עמוד פרטו על שלושה בעלי חיים העונים להגדרת העמוד.

5. הוסיפו בעמוד הראשי 3 קישורים, מקמו את הקישורים בתוך פסקה על מנת שכל קישור יופיע בשורה נפרדת: לבעלי החיים טורפים, לעופות, לבעלי החיים הימיים, כל קישור יקשר לדף המתאים.

6. תחת כל קישור, הוסיפו שורת קישורים (בשורה אחת, לא בתוך התגית p) לשלושה בעלי חיים שפרטתם בסעיף 4, כך קישור יקפיץ למיקום הפירוט על בעל החיים המתואר.

7. בכל עמוד צרו קישור חזרה לעמוד הראשי.

אופן פתיחת הקישור

באופן כללי קישור נפתח בחלון הנוכחי בו אנו צופים, אך ניתן גם להגדיר שהקישור יפתח, בלשונית חדשה, משתמשים באופציה כזו כאשר משלבים קישור בדף, קישור לאתר מקור, או קישור לאתרים שונים, אך מעוניינים שהגולש לא יצא מהאתר, כאשר ילחץ על הקישור, אלא רק יעיין בקישור המובא.

להגדרת הקישור שיפתח בלשונית חדשה נשתמש במאפיין target,

כאשר נמקם במאפיין target את הערך _blank, הגדרנו לדפדפן, שהקישור יפתח בלשונית חדשה.

לדוגמא:

```
<a href="https://www.gov.il/" target="_blank">השירותים</a>
```

קישור זה מגדיר שבעת לחיצה על הקישור, הכתובת <https://www.gov.il> תפתח בלשונית חדשה.

ניתן גם להגדיר לקישור שלא יוביל למקום כלשהו, על ידי סימון הסימן # ב-href, נשתמש בצורת הגדרה כזו כאשר ניצור קישורים לכל האתר, בכל דף הקישור מתוך רשימת הקישורים המתייחס אליו עצמו יקושר כך.

לדוגמא:

```
<a href="#" >דף הבית </a>
```

1. צרו מסמך html חדש.
2. הוסיפו למסמך קישורים ל-4 אתרים.
3. הגדירו שהקישור יפתח בלשונית חדשה.

כתובת יחסית וכתובת מוחלטת

יקל עלינו להבין את ההבדל בין כתובת יחסית לכתובת מוחלטת, אם נמשיל את האתר שלנו לכיתה, כשנרצה לקרוא למשהו מהכיתה, נקרא לו בשמו ומובן שקראנו לבעל אותו שם מתוך חדר הכיתה, אך אם נרצה לקרוא לתלמיד אחר מרחבי בית הספר - מחוץ לכיתה, נפרט את שמו ואת הכתובת המלאה שלו - הכיתה שבה הוא לומד מיקום הכיתה וכדו', כך ממש גם בכתובת יחסית וכתובת מוחלטת באתר.

כתובת url מוחלטת (Absolute address), היא כתובת מלאה הכוללת את כל חלקי הכתובת: הפרוטוקול תקשורת עם האתר (http או https), את כתובת השרת (דומיין-תחום או כתובת ip) ואת כתובת המשאב (מסמך html לדוגמא).

היא מודיעה לנו במדויק היכן האתר יושב, עם כל כתובתו "המלאה".

לעומת זאת כתובת יחסית (Relatively address), היא כתובת שנשתמש בה לשם קישור לדפים היושבים באותו אתר בו מתבצעת הקישור, היא לא מפרטת את כל אופן התקשורת עם האתר, אלא רק מודיעה את שמו, במידה ויושבת ומופיעה באותה רמה כשל הדף הקורא לה.

לדוגמא, אם יש לנו דף "ראשי" שמקושר לדף "אודות", דף "צור קשר" ודף "מפעילויות הארגון". אם הדף הראשי ממוקם באותה תיקייה בה ממוקמים שאר הדפים, בעת קישור הדף הראשי עם שאר הדפים, אין צורך להתייחס לכל הכתובת (פרוטוקול התקשורת, ודומיין), אלא מספיק לקרוא לדף בשמו + סיומת סוג הקובץ (php, html, וכדו').

1. צרו מסמך html חדש.
2. צרו שני קישורים לכתובות הבאים:

(1) <https://www.gov.il>

www.gov.il (2)

3. צפו במסמך בדפדפן.

4. אילו מבין הקישורים מוביל לאתר השירותים הממשלתיים?

5. מה קורה כאשר לוחצים על קישור מספר 2?

קישור לחיץ לטלפון ולכתובת מייל

מספר טלפון לחיץ באתר

באמצעות קישור ניתן להפוך את מספר הטלפון המופיע באתר למספר לחיץ, כך שהגולש יוכל ללחוץ עליו והנייד שלו יבצע חיוג.

בכדי להפוך קישור ללחצן "מחייג" נכניס את מספר הטלפון שלנו או כל טקסט אחר שיוצג לגולש בתוך התגית `<a>`, ובמאפיין `href`, נכניס את ערך מספר הטלפון שלנו לאחר המילה `tel:`, לדוגמא כך:

`צלצלו אלינו`

לאחר שהגדרנו הגדרה זו לחיצה על הקישור "צלצלו אלינו" תוביל לחיוג מהנייד למספר הטלפון 050-0000000.

אם צפוי שחלק מהמצלצלים יבצעו את השיחה גם מחוץ לאזורי הארץ, ניתן להוסיף קידומת של הארץ לדוגמא:

`צלצלו אלינו`

כתובת מייל כקישור פעיל

ניתן להפוך את כתובת המייל באתר ל"כתובת לחיצה", כך שיפתח חלון חדש לכתובת מייל לכתובת המייל שלנו (במידה ומותקנת על מחשב הלקוח תוכנת דוא"ל).

לשם כך ניצור טקסט לשליחת אימייל ממוקם בתוך התגית `<a>` ובמאפיין `href`: נכניס את הערך `mailto:` בתוספת כתובת המייל שלנו.

לדוגמא:

`` תרגישו חופשי לשלוח מייל ``

מעבר לפתיחה האוטומטית של טופס האימייל, נוכל לגרום גם למרכיביו השונים של הודעת דואר אלקטרוני סטנדרטי לקבל ערכים. למשל הכנסת כתובת לתיבה "עותק" שבהודעת

דואר אלקטרוני, או הכנסת כתובת לתיבה "עותק מוסתר", אף ניתן להכניס טקסט לנושא הודעת הדואר, וכמה מילים בגוף הודעת הדואר.

המבנה המלא הוא כזה:

<a

href="mailto:mail@simple.com?cc=copymail@simple.com&bcc=hidecopy@simple.com&subject=mailto subject&body= זהו גוף ההודעה

>שלח הודעה".במייל

לאחר הגדרת הכתובת שלנו, המוגדרת מיד לאחר המילים mailto שהיא הכתובת למשלוח דואר, מופיע סימן שאלה, ולאחריו המאפיינים המוגדרים:

Cc - לאיזו כתובת ישלח עותק.

Bcc - לאיזו כתובת ישלח עותק מוסתר.

Subject - מה יהיה נושא הודעת הדואר.

Body - טקסט בגוף ההודעה.

כשבין מאפיין למאפיין מופיע הסימן &, לשרשור כל הקוד לקוד אחד.

לדוגמא עבור הפרמטרים הבאים:

נושא: אני רוצה לדבר אתכם הגעתי דרך האתר

שלח ל: mail@simple.com

שלח העתקים ל: copymail@simple.com

שלח העתקים נסתרים ל: hidecopymail@simple.com

גוף ההודעה: רציתי לברר על השירות שאתם נותנים, על מנת לרכוש ממכם את אחד ממוצריכם.

נכתוב את הקישור הפעיל בצורה כזו:

<a

href="mailto:mail@simple.com?cc=copymail@simple.com&bcc=hideco

pymail@simple.com&subject= הגעתי דרך, אתכם לדבר אני רוצה

רציתי לברר על השירות שאתם נותנים, על מנת לרכוש ממכם את אחד &body=האתר

>שלח הודעה".ממוצריכם

תרגיל 12

1. צרו מסך html חדש
2. הוסיפו קישור המאפשר "לחייג אליך". הטקסט היוצג לגולש הוא: "חייגו אלי"
3. הוסיפו קישור המאפר לשלוח אליך מייל, הכינו בקישור את נושא ההודעה ותוכן ההודעה.

הוספת תמונות באתר

תגית - תגית תמונה

כדי להוסיף תמונות למסמך נשתמש בתגית , תגית ה-`` מורכבת מתגית בודדת פותחת ואין לה תגית סגירה, לתגית ה-``, ארבעה מאפיינים שנגדיר, על מנת להגדיר היכן נמצאת התמונה שנרצה להציג, ומהם התכונות שנרצה שהתמונה תקבל: רוחב גובה וטקסט חילופי.

מאפייני תגית תמונה:

1. **Src** - במאפיין זה נציין את הנתוב המלא או היחסי לקובץ התמונה באתר.
2. **Whidth** - במאפיין זה נגדיר את רוחב התמונה בפיקסלים.
3. **Height** - במאפיין זה נגדיר את גובה התמונה בפיקסלים.
4. **Alt** - (alternative - חלופה), במאפיין זה נגדיר טקסט חילופי, טקסט המתאר את התמונה, טקסט זה לא יופיע במסמך ה-`html`, אלא מיועד להגדיר לקוראי מסך, מהו תוכן התמונה, מאפיין זה חשוב מבחינת קידום האתר, וחשוב לנגישות האתר לבעלי מוגבלויות ראייה, הנעזרים בקוראי מסך שונים, במידה והתמונה לא נמצאה על ידי הדפדפן להצגה או הדפדפן לא הצליח להציג את התמונה מסיבות של חיבור איטי וכדו', יוצג טקסט זה כחלופה לתמונה שהייתה צריכה להופיע.

לדוגמא:

```

```

חשוב מאד: לשמור על פרופורציות ולא להגזים בנפח התמונות המשולבות בדפי `html`, על מנת שדף `html`, יוכל להטען מהר ככל האפשר, עלינו "להקל" עליו, וכשאנחנו מוסיפים אלמנטים כעין תמונה או כל מדיה שהיא, נבדוק איך נוכל לצמצם ככל האפשר במשקל הקובץ.

אם התמונות שלנו "כבדות" מידי קיימים ברשת מספר אתרים המיועדים להקטנת נפח תמונות, ניתן להיעזר בהם.

הנה כמה מהם שעלו בחיפוש בגוגל:

<https://www.easy-resize.com/he/>

<https://compressjpeg.com/>

<https://kraken.io/web-interface>

<https://tinypng.com/>

תמונות הממוקמות בתיקייה שונה

לצורך ארגון האתר, מומלץ לחלק את חומרי המדיה המוצגים באתר לתיקיות לפי נושאים ועניינים.

מקובל להכין תיקייה בשם images וכדו' שבתוכה מאחסנים את תוכן התמונות של האתר. כיון שקובצי דפי האתר ממוקמים בתיקייה שונה עלינו להתייחס לכך במאפיין src, המציין את הנתיב לקובץ.

ניצור את המסלול מהדף בו אנו רוצים להציג תמונה למיקום התמונה, אם יש צורך לעלות תיקייה נשתמש בסימן נקודה נקודה ולוכסן כך:..

אם יש צורך להיכנס לתיקייה אחרת נכנס כך: **namefolder/**

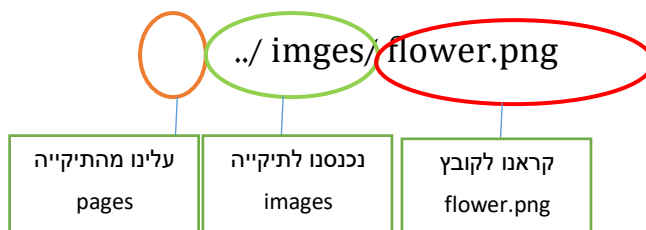
לדוגמא אם נרצה להיכנס לתיקייה img, נכנס אליה כך: **img/**

דוגמא מורכבת יותר: הדף "אודות" ממוקם בתיקייה הנקראת pages, במקביל לתיקייה זו קיימת תיקייה הנקראת images, תיקייה המכילה את תמונות האתר, כשנרצה להציג את הקובץ flower.png, מתוך התיקייה images על גבי הדף "אודות" נשתמש בקישור כך:

```

```

מה עשינו בנתיב המפנה לקובץ flower?



תרגול תרגיל 13

1. צרו דף html חדש.

2. הוסיפו בדף כותרת "ברוכים הבאים לירקניה"
3. שמרו את הדף.
4. צרו 3 דפים נוספים: פירות, ירקות, עלים.
5. הוסיפו לכל עמוד לפחות 4 תמונות מתאימות.
6. הקפידו להגדרה מתאימה במאפיין alt בכל תמונה.
7. קבעו לתמונות רוחב וגובה שיראו טוב בדף.
8. את כל התמונות מקמו בתיקייה בשם images.
9. חלקו את התיקייה לתתי תיקיות: תיקיית fruits לאחסון תמונות הפירות, תיקיית vegetables לאחסון תמונות הירקות ותיקיית Leaves לאחסון תמונות העלים.
10. הקטינו את נפח התמונות ככל האפשר.
11. הוסיפו קישור מכל עמוד לכל עמוד (סה"כ 4 עמודים).

תמונה כקישור

ניתן ליצור תמונה שתהווה קישור למיקום כלשהו באתר או מחוצה לו, על מנת שתמונה תהפוך לקישור יש למקם את תגית ה-`` בתוך התגית `<a>`.
לדוגמא:

```
<a href="home.html"></a>
```

בדוגמא זו מיקמנו תמונת בית כקישור לדף הבית.

תרגיל נוסף 14

1. צרו מסמך html חדש.
2. הוסיפו במסמך 3 תמונות של כלי מטבח: כוס, מחבת וסיר, סכום.
3. שמרו את הדף בשם index.html.
4. צרו 3 דפים:
 - a. דף מספר 1:
 - i. הוסיפו בדף כותרת: "כוסות"
 - ii. הוסיפו 5 תמונות של כוסות שונים.
 - b. דף מספר 2:

- i.** הוסיפו בדף כותרת: "מחבתות"
- ii.** הוסיפו כ-5 תמונות של מחבתות.
- c.** דף מספר 3:
- i.** הוסיפו בדף כותרת "סירים"
- ii.** הוסיפו כ-5 תמונות של סירים שונים.
- d.** דף מספר 4:
- i.** הוסיפו בדף כותרת "סכום"
- ii.** הוסיפו כ-5 תמונות של פרטי סכו"ם שונים.
- 5.** קבעו שכל תמונה בעמוד הראשון תקשר לדף המתאים.
- 6.** בכל התמונות הקפידו להגדרה מתאימה במאפיין alt בכל תמונה.
- 7.** קבעו לתמונות רוחב וגובה שהתמונות תראינה טוב באתר.

רשימות

רשימות

קיימות שלשה סוגי רשימות:

1. **רשימה מסודרת** - רשימת פרטים, כשכל שורה ממוספרת באמצעות מספרים, אותיות, או ספרות רומיות.
2. **רשימה לא מסודרת** - רשימת פרטים, כשכל שורה מסומנת באמצעות תבליט.
3. **רשימת הגדרות ומונחים** - רשימה שאינה ממוספרת ואין לה תבליטים.

רשימה מסודרת - התגית

בכדי ליצור רשימה מסודרת נשתמש בתג . בכדי להכניס פריטים לרשימה, נשתמש בתג .

דוגמא לרשימה מסודרת:

```
<h2>רשימת פירות</h2>
```

```
<ol>
```

```
<li>תפוח</li>
```

```
<li>ענבים</li>
```

```
<li>אגס</li>
```

```
<li>משמש</li>
```

```
<li>אפרסק</li>
```

```
</ol>
```

תצוגת הקוד בדפדפן



ניתן להגדיר את סגנון צורת סידור הרשימה באמצעות המאפיין type, מאפיין זה יכול להכיל את אחד מהערכים:

1 - לרשימה ממסופרת עם מספרים.

A - לרשימה ממסופרת עם אותיות גדולות באנגלית.

a - לרשימה ממסופרת עם אותיות קטנות באנגלית.

I - לרשימה ממסופרת באמצעות ספרות רומיות גדולות.

i - לרשימה ממסופרת באמצעות ספרות רומיות קטנות.

את המאפיין `type`, נגדיר בתגית ``, כך:

```
<ol type="1">
```

ניתן גם להגדיר ערך התחלתי לרשימה, במידה והרשימה אינה מתחילה ב1.

נגדיר ערך התחלתי באמצעות המאפיין `start` כך:

```
<ol start="4">
```

תרגיל 15

1. צרו מסמך html חדש.

2. הוסיפו כותרת "מכוניות"

3. הרשימה תכיל את רשימת המכוניות הבאה:

1. סוזוקי

2. סיאט

3. שברולט

4. פיג'ו

5. סובארו

6. פורד

7. פולקסווגן

8. פיאט

4. הרשימה תהיה ממסופרת באמצעות ספרות רומיות.

רשימה לא מסודרת - התגית

בכדי ליצור רשימה לא מסודרת יש להשתמש בתג , להתחלת הרשימה, בכדי להכניס פריט לרשימה, נשתמש בתג .

דוגמא לרשימה:

```
<ul>
```

```
<li>אחת</li>
```


```
<li>שתיים</li>
```

```
<li>שלוש</li>
```

```
<li>ארבע</li>
```

```
</ul>
```

תצוגת הקוד בדפדפן



- אחת
- שתיים
- שלוש
- ארבע

ניתן להגדיר את סגנון צורת סימון הפריט באמצעות המאפיין `type`, מאפיין זה יכול להכיל את אחד מהערכים:

circle - עיגול חלול לצד כל פריט ברשימה.

disc - עיגול מלא לצד כל פריט ברשימה, אם לא נגדיר את המאפיין `type`, תבליט זה יבוא כברירת מחדל לצד כל שורה.

square - ריבוע קטן לצד כל פריט ברשימה.

none - רשימה ללא סימון.

את המאפיין `type`, נגדיר בתגית , כך:

```
<ul type="Square">
```

תרגיל 16

1. צרו מסמך html חדש.
2. הוסיפו כותרת "מכשירי כתיבה".
3. צרו רשימה בלתי ממוספרת, הרשימה תכיל את הפרטים הבאים:
 - מחשבון
 - עיפרון

- מחדד
- מחק
- עט
- צבעים
- סרגל

4. לצד כל פריט ברשימה יופיע ריבוע קטן.

רשימת הגדרות ומונחים - התגית <dl>

רשימה זו היא רשימה של מונחים, עם תיאור עבור כל מונח.

התגית <dl> מגדירה את הרשימה, התגית <dt> מגדירה את המונח, התגית <dd>, מתארת את המונח.

לדוגמא:

<dl>

<dt>קפה</dt>

<dd>משקה שחור</dd>

<dt>חלב</dt>

<dd>משקה לבן</dd>

<dt>סודה</dt>

<dd>משקה מים מוגז</dd>

</dl>

תצוגת הקוד בדפדפן	
קפה	משקה שחור
חלב	משקה לבן
סודה	משקה מים מוגז

תרגול רשימת הגדרות - תרגיל 17

צרו מסמך html חדש.

צרו במסך רשימה לתיאור והגדרת המונחים הבאים:

שמחה

מקרין אושר ומצב רוח טוב

עצב

תחושה או מצב רוח לא נעימים

חכם

בעל יכולת שכלית מפותחת, בעל ידע

מהיר

בעל יכולת לנוע בקצב גבוה

קינון רשימות

ניתן לקנן רשימות אחת בתוך השניה, כך שתהיה רשימה בתוך רשימה, בתגית `` מסוימת תהיה תגית פותחת של רשימה חדשה `` או ``, עם תוכן רשימתי (תגיות `li` ברשימה).

לדוגמא:

`<h2>רשימה מדורגת</h2>`

``

``

פירות

``

`תפוח`

`ענבים`

`אגס`

``

``

``

ירקות

``

`מלפפון`

`עגבניה`

`פלפל`

``

``

תצוגת הקוד בדפדפן

רשימה מדורגת

- פירות
 - תפוח
 - ענבים
 - אגס
- ירקות
 - מלפפון
 - עגבניה
 - פלפל
- עלים
 - חסה
 - כוסברה
 - נענע

```

<li>
  עלים
  <ul>
    <li>חסה</li>
    <li>כוסברה</li>
    <li>נענע</li>
  </ul>
</li>
</ul>

```

תרגול רשימה בלתי ממוספרת - תרגיל 18

1. צרו מסמך html חדש.
2. הוסיפו כותרת "משקאות":
3. צרו בדף ה-html את הרשימה הבאה:
 - משקאות חמים:
 - תה
 - קפה
 - קפוצ'ינו
 - סחלב
 - משקאות קלים:
 - מיץ תפוזים
 - מיץ אשכוליות
 - מיץ מנגו
 - משקאות מוגזים:
 - קולה
 - לימון ליים
 - מירינדה
 - משקאות חריפים

- יין
- וויסקי
- קוניאק
- ליקר
- וודקה

תרגול רשימה מקוננת - תרגיל 19

1. צרו מסמך html חדש.
2. הוסיפו כותרת "כלי תחבורה":
3. צרו בדף ה-html את הרשימה הבאה:

- באוויר
 - מטוס
 - מסוק
 - חללית
 - כדור פורח
- בים
 - אנייה
 - סירה
 - יאכטה
- ביבשה
 - אופניים
 - אופנוע
 - מכונית
 - משאית

עיצוב גליון בעזרת CSS, שילוב CSS וסלקטורים

עיצוב גליון html בעזרת CSS

תגי ה- html, הם אבני הבניין בבניין דפי האינטרנט, ע"מ לעצב את הדף יש להשתמש בהגדרות עיצוביות, לכל אלמנט במסמך ה- html ביכולתנו להגדיר עיצוב ויזואלי כמו צבעים, גוונים, רקעים, סמן העכבר, קווים, מסגרות, משקלי גופנים, סוגי גופנים, מרווחים ועוד ועוד...

לדוגמא קוד ה- html הבא:

```
<!DOCTYPE html>

<html>

<head>

  <title>עיצוב דף</title>

</head>

<body dir="rtl">

  <p>זו פסקה אחת</p>

  <p>זו פסקה שניה</p>

  <p>זו פסקה שלישית</p>

</body>

</html>
```



נוסיף לפסקאות שבו את פקודת ה- CSS הבאה:

```
p {  
  color: red;  
}
```



והתוצאה שנקבל היא כזו:

בשנת 1995 קבוצת התקינה w3c, יצרה את ה-css, לפני כן נכתבו האתרים בשילוב התוכן וסגנונות עיצוביהם באותו מסמך, מסמך ה-html, עבודה בצורה כזו הייתה עבודה מסורבלת, והפכה את דפי ה-html למפתחים כמסורבלים ובלתי קריאים, כשנדרש שינוי לאלמנטים באתר, היה צורך לפתוח את דף ה-html לעבור אלמנט, אלמנט בדף ולשנות בו את ערכיו העיצוביים בצורה ידנית, מה שלקח זמן רב, המצאת ה-css מאפשרת לקבץ את כל ההגדרות העיצוביות למסמך אחד, ומשרתת מספר רב של דפים ואלמנטים מכל חלקי האתר, כך שלכל שינוי עיצובי יש לגשת אך ורק לקובץ ההגדרות העיצוביות, ושינוי קל במסמך ה-css יכול להשפיע בו זמנית על מאות אלמנטים באתר, לדוגמא נוכל להגדיר שינוי של צבע הפסקאות, כלומר שינוי צבע של התגית <p>, לצבע חדש, וברגע קל כל הפסקאות תקבלנה את הגדרת הצבע החדשה שהגדרנו, בנוסף מהמצאת ה-css קוד דף ה-html נעשה לפחות עמוס ונקי יותר, יתרון נוסף ל-css, הוא האפשרות להתאים את האתר להדפסה, לתצוגה רספונסיבית - מותאמת לשלל גדלי מסך.

מהי CSS

Css ראשי התיבות של המילים **Cascading Style Sheet** - ובעברית גליונות סגנון מדורגים, css הינו פורמט לעיצוב דפי אינטרנט לקביעת העיצוב של התגים בhtml.

שלוש שיטות לשילוב גליון CSS

שלושה דרכים לשלב CSS במסמך HTML:

1. **External CSS** - (CSS חיצוני) קובץ נפרד חיצוני, קובץ הנשמר בסיומת CSS.

הדרך המומלצת והפשוטה ביותר לשילוב קוד CSS באתר, היא ליצור קובץ CSS חיצוני וליצור קישור לקובץ, מדפי ה-HTML שיושפעו מהקובץ.

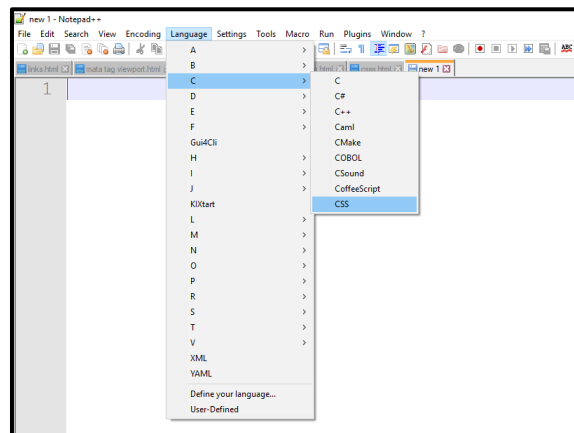
קובץ CSS חיצוני, כמו קובץ HTML ניתן ליצור באמצעות כל עורך טקסט, סיומת הקובץ היא: CSS.

יצירת מסמך CSS

1. פתחו את תכנת ה notepad++.

2. הגדירו שאתם עומדים לכתוב דף CSS כך:

בתפריט גשו Language (שפה) < האות C < CSS.



שמירת מסמך CSS

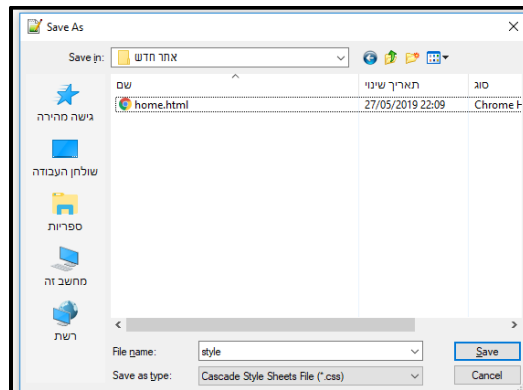
1. נשמור את הקובץ בתיקיית האתר שאנו מעוניינים לקשר אליו את הגדרות ה-CSS שנכתוב בקובץ.

2. בתפריט נבחר `file> save as...`

3. ננתב למיקום התיקייה שלנו, ששם נשמור את הקובץ.

4. בתיבה `file name`, נקליד את שם הקובץ. (שם הקובץ צריך להיות באנגלית).

5. וכמובן נלחץ על save, לשמירת הקובץ.



שימוש בקובץ CSS השמור



לאחר השמירה, נקבל מסמך שנראה בתיקיה כך:

נקשר את קובץ ה-CSS לדף ה-HTML, את הקישור ניצור בדף ה-HTML שמקבל את הגדרות ה-CSS ונמקמו בתוך האלמנט <head> על ידי ההגדרה הבאה:

```
<head>
```

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

```
</head>
```

את הקישור יש למקם בכל דף ודף בנפרד.

2. **Internal css** - (CSS פנימי) ניתן לשלב סגנון ה-CSS באמצעות מיקום משפטי

ה-CSS בראש דף האינטרנט, דף ה-HTML, בתוך התגית <style>, את תגית ה-style

נמקם בתוך התגית head של המסמך, לדוגמא:

```
<head>
```

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

```
    P {
```

```
        Color: red;
```

```
        Text-align: center;
```

```
    }
```

</style>

</head>

דרך זו לא מומלצת, כיוון שאם נרצה לקבוע שינוי הנוגע לכל חלקי האתר, במקום לשנות שינוי במקום אחד נצטרך לגשת לכל דפי האתר ולשנות בכל קובץ וקובץ בנפרד, ניתן להשתמש בה ל"בדיקה" זמנית, או שילוב של משהו זמני.

3. **Inline css** - (בשורה) **משולב בתוך תגיות ה- html**

שיטה זו היא השיטה הפחות מומלצת מבין שתי השיטות הקודמות שילוב של קוד ה-CSS בתוך האלמנט אליו מתייחסת ההגדרה, על ידי הוספת המאפיין style, כשהערכים שהוא מקבל הם שם המאפיין ב-CSS וההגדרה לדוגמא:

<p style="color: red"> בפסקה זו הטקסט הוא בצבע אדום </p>

דרך זו לא מומלצת כלל, היא יוצרת כפילויות רבות של קוד, על ידי הגדרות ערכים זהים למספר רב של אלמנטים, מה שיכולנו לחסוך בשתי הדרכים הקודמות.

תרגיל 20

1. העתיקו את התיקייה: שילוב CSS בשלש דרכים לתיקייתכם האישית.
2. פתחו מתוכה את התיקייה שילוב CSS בתגית ה-head.
3. פתחו את הקובץ `Css integration.html` לתצוגה בדפדפן.
4. פתחו את הקובץ `Css integration.html` לעריכה בעורך הטקסט.
5. העתיקו את הגדרות ה-CSS הבאות לתגית style בתוך תגית ה-head:

```
H1{ color: blue;}
```

```
P{ color: orange; }
```

6. שמרו את השינויים בקובץ.
7. רעננו את הדף על מנת לצפות בשינויים שחלו אודות לשילוב ה-CSS. (באמצעות המקש f5 או צירוף המקשים ctrl+r).
8. פתחו את התיקייה שילוב CSS בגוף התגית.
9. פתחו את הקובץ `Css integration.html`.

10. שלבו את שורת ה-css המתחילה ב-h1 בתגית h1.
11. שלבו את שורת ה-css המתחילה ב-p בתגיות p.
12. שמרו את השינויים.
13. פתחו את הקובץ בדפדפן וצפו בתוצאות.
14. פתחו את התיקייה שילוב css בקובץ נפרד.
15. צרו מסמך css חדש, העתיקו לתוכו את הגדרות ה-css מסעיף 4.
16. שמרו את המסמך בשם: style.css, בתיקייה 'שילוב css בקובץ נפרד'.
17. פתחו את הקובץ בדפדפן וצפו בתוצאות.

תרגיל 21

1. פתחו שוב את התיקייה שילוב css בשלוש דרכים.
2. שנו את ההגדרה:

```
P{ color: orange; }
```

להגדרה:

```
P{ color: blue; }
```

3. החילו את השינוי בכל אחד מהקבצים, כך שהפסקאות במסמך ה-html תשתנינה צבען מכתום לכחול.

התנגשויות ברמות שונות של CSS

מה קורה אם השתמשנו בשלושת השיטות להגדרת ה-css ובכל אחד מהמקומות הגדרנו לאותו אובייקט הגדרה שונה?

לדוגמא, הגדרנו לתגית p ב-css החיצוני צבע ירוק, ב-css הפנימי, בתוך תגית ה-style, צבע אדום, ובשורת האלמנט, הגדרנו בתוך התגית של ה-p צבע כתום.

איזה צבע הפסקה תקבל?

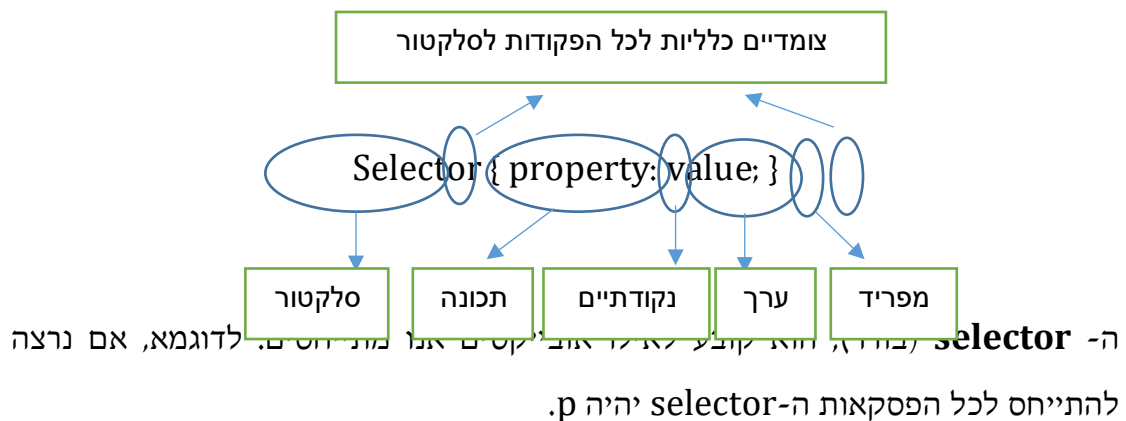
הדפדפן מתייחס לפי סדר חשיבות בין השיטות, כך שאם הוא מגיע לקונפליקט שכזה הוא פועל לפי סדר העדיפויות, מהעיקרי לטפל:

1. **Inline css** - CSS המשולב בתוך האלמנט, גובר בחשיבותו.
2. **Internal css** - סגנון המוגדר בראש העמוד, בתגית `style`, ב-`head` של העמוד. שני בחשיבותו.
3. **External css** - CSS היושב בקובץ חיצוני, שלישי בחשיבותו.
4. **browser default** - הגדרות ה-CSS ברירת מחדל של הדפדפן, הפחותות ביותר בחשיבותן.

כך שלשאלתנו הפסקאות באתר שהוגדרו בהם את שלושת השיטות לאותו האלמנט, כפי התיאור תקבלנה את הצבע שהוגדר בשילוב ה-`html`, בשורת האלמנט, הצבע הכתום.

מבנה בסיסי של פקודת CSS

להגדרות ה-CSS מבנה קבוע:



{ } צמדים פותחים לנו וסוגרים לנו את בלוק ההוראות לאותו סלקטור.

ה- **property (תכונה)** קובע איזה סוג תכונה אנו מגדירים כגון: גודל, צבע, רוחב, גובה וכדו', לאחר שם התכונה יבואו נקודותיים:

ה- **value (ערך)** - קובע מה הערך שנרצה להגדיר לתכונה, לדוגמא לתכונה של צבע, התכונה `color`, הערכים האפשריים, הם צבעים שונים, בצורות כתיבה שונות כגון, `red`, `green` או `#ca21ac` וכדו'.

בכדי לתת מספר הגדרות עיצוביות לאותו אלמנט, ניתן להניח לסלקטור אחד מספר רב של הגדרות, הבניות מצמידים של תכונות וערכים, לסיום זוג של תכונה וערך והפרדה מזוג התכונות הבא נניח את הסימן ; -פסיק נקודה.

דוגמא:

```
h1 { color:blue; font-size:12px;}
```

הגדרנו סלקטור שנקרא h1, כלומר משפיע על כל הכותרות הראשיות באתר, וקובע את ההגדרות הבאות, הגדרת color - צבע, לצבע כחול, הגדרת font-size - גודל פונט ל- 12 פיקסלים.

הוספת הערה בקובץ CSS

ניתן להוסיף הערות על מנת להסביר את קוד ה-CSS, הדפדפן מתעלם מהתוכן המוצג בתוך ההערה, להוספת הערה בקובץ ה-CSS נתחיל את כתיבת ההערה בסימן /* - לוכסן ו*- כוכבית, ונסיים את ההערה בסימן *-כוכבית ולוכסן, כך:

```
P {
```

```
    Color: red;
```

```
    /* למרכז הטקסט */
```

```
    Text-align: center;
```

```
}
```

```
/* זו הערה ארוכה */
```

```
/* המתפרסת גם על יותר משורה אחת
```

תרגיל 22

1. פתחו את הקובץ newstyle.css מהתיקייה שיעור 7.
2. הוסיפו בקובץ את ההערה הבאה: קובץ זה מוסיף צבע רקע צהוב למסמך, צובע את הכותרות בצבע כחול ואת הטקסטים בצבע סגול.

הורשת תכונות

רב הסלקטורים בשפת CSS, תומכים בהורשת תכונות לסלקטורים נוספים המקוננים בתוכם. כך שאם נגדיר למשל צבע טקסט חום לסלקטור `body`, כל הפסקאות והכותרות היושבות בתוך ה-`body` תקבלנה צבע חום, אלא אם כן נגדיר אחרת לאותם אובייקטים.

יחידות מידה בCSS

יש תכונות שערכיהם יוגדרו בערך מספרים, ונצטרך לציין את יחידת המידה עבור הערך. היחידות מידה הנפוצות הן:

Px - לציין פיקסלים לדוגמא: `font-size:12px`

Rem - יחידת מידה המחושבת לפי גודל פונט הברירת מחדל של אלמנט ה-`html`. כברירת מחדל, גודל פונט במסמך `html` הוא `16px` ולכן `1rem` הוא `16px`.

שימוש לדוגמא: `font-size: 1rem;`

em - יחידת מידה המחושבת לפי גודל הפונט המוגדר באלמנט האב, לדוג' אם קבענו ערך `2em` הפונט יגדל פי 2 מהגודל שקבענו באלמנט הראשי.

Pt - גודל בנקודות, לשימוש בד"כ לדפוסל לדוג' `font-size:12pt`.

% - גודל באחוזים, לדוג' `width:80%`, מגדיר לאלמנט גודל של 80% מרוחב המסך, הגודל משתנה בהתאם לגודל המסך.

Vw - (`viewport width`) יחידת מידה המייצגת את רוחב החלון המוצג, הגדרה של `100vw` תגרום לאלמנט לקבל רוחב מלא, לכל רוחב החלון.

Vh - (`viewport height`) יחידת מידה המייצגת את גובה החלון המוצג, הגדרה של `10vh` תגרום לאלמנט לקבל את מלא גובה החלון.

הערה: כאשר הערך הוא 0, אין צורך לציין את יחידת המידה.

Selectors - בוררים

לפי אלמנט

סלקטור לפי תגית, סלקטור כזה מבוסס על שם האלמנט, התגית, לדוגמא: p או h1 או a וכדו'.

לדוגמא: הקוד הבא ישפיע על כל מקום שמופיע התגית h1 ויהפוך את הטקסט שבו לאדום:

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

לפי id - זיהוי ייחודי

בהזדמנויות מסוימות נרצה שאלמנט מסוים מקבוצת אלמנטים יקבל עיצוב ייחודי ושונה משאר האלמנטים מאותו סוג, כדי לבחור אלמנט מסוים, ספציפי, נשתמש במזהה, את המזהה נוסיף לאלמנט בדף ה-html באמצעות המאפיין id, כשהערך של ה-id הוא המזהה הספציפי לאלמנט. לדוגמא:

```
<p id="one" >
```

```
<p id="two" >
```

על מנת להגדיר הגדרות CSS לאלמנטים שהגדרנו להם id נשתמש ב-id שהגדרנו בתוספת הסימן # לפני שם ה-id.

לדוגמא:

```
#one { color: red; }
```

```
#two { color: blue; }
```

לאחר הגדרות אלו, הפסקה הראשונה תקבל צבע אדום לטקסט, ואילו הפסקה השנייה תקבל צבע כחול.

שימו לב: ה-id הוא ייחודי, ולכן לא יכולים להופיע מספר אלמנטים בדף בעלי id זהה.

לפי class

נוכחנו שסלקטור לפי id מאפשר לבחור אלמנט לפי המזהה הייחודי שהגדרנו לו ולפיו לקבוע סגנון, זו שיטה מעולה כאשר קיים אלמנט בודד שנרצה להחיל עליו סגנון ספציפי, שאינו משמש אלמנטים נוספים, אך אם יש לנו מספר פסקאות מתוך שלל הפסקאות שנצרה לתת להן הגדרת עיצוב שונה משאר הפסקאות באתר, אם נרצה להגדיר עיצוב CSS לקבוצת אלמנטים נגדיר סלקטור מחלקתי, סלקטור לפי class.

לדוגמא אם נרצה שכל הפסקאות תהינה בצבע כחול, אך פסקאות שמכילות דוגמת הסבר תצבענה בצבע אפור, נוסיף לתגית p בפסקאות הדוגמא את קלאס משותף, נניח קלאס בשם example את הקלאס נגדיר באמצעות המאפיין class, כשהערך שניתן לו הוא שם הקלאס שקבענו.

לדוגמא:

```
<p class=example>
```

על מנת להגדיר הגדרות CSS לאלמנטים שהגדרנו להם class מסויים, נשתמש ב-class שהגדרנו בתוספת הסימן. (נקודה) לפני שם class.

לדוגמא:

```
. example { color: gray; }
```

שימו לב: כל אלמנט יכול להכיל מזהה ייחודי ובנוסף מספר קלאסים שונים.

לדוגמא:

```
<p id="new" class=" example checked ">
```

נשים לב על מנת להגדיר מספר קלאסים לאותו אלמנט השתמשנו בתו רווח " " בין קלאס אחד למשנהו, הפסקה הנ"ל תקבל את ההגדרות שמשמשות בשלל הסלקטורים שלה.

```
. example {  
  color: gray;  
}
```

```
. checked {
  background-color: yellow;
}
```

```
#nee {
  font-weight: bold;
}
```

לפי attribute

סלקטור מסוג attribute מאפשר לבחור אלמנטים המכילים מאפיינים מסוימים ולהכיל עליהם את הגדרות ה-CSS, להגדרת הסלקטור נשתמש במאפיין ובערך, כך שיופיע שם המאפיין, הסימן שווה, והערך כשהם ממוקמים בתוך סוגריים מרובעות לדוגמא אם נרצה לבחור את כל האלמנטים שמוגדר להם מאפיין שפה: אנגלית ולצבוע אותם בצבע אפור נעשה זאת כך:

```
[lang=en] { color: gray; }
```

או אם נרצה להגדיר שכל הפסקאות שהוגדר להם title, יקבלו צבע אדום, נעשה זאת כך:

```
P[title] {color: red;}
```

היחס של המאפיין וההגדרה לא חייב להיות בייחס של שווה ל-, ניתן גם למדוד את היחס באופן הפוך, לא שווה ל, כך:

```
P[class~example]{ color: green; }
```

הגדרנו שכל הפסקאות שאין להם קלאס בשם example תוגדרנה בצבע ירוק.

הסלקטור *

כוכבית מציינת **הכל** - כל האלמנטים, הסלקטור כוכבית, קובע שכל האלמנטים באתר יקבלו את הגדרת ה-CSS שתוגדר.

לדוגמא:

```
* { background-color: yellow; }
```

הגדרה זו תגדיר שכל סוגי האלמנטים באתר יקבלו רקע צהוב.

קבוצת סלקטורים

אם קיימים מספר אלמנטים שנרצה לתת להן הגדרות CSS זהות, לא נתן לכל אלמנט את ההגדרות מחדש, אלא ננסה למזער את הקוד ככל האפשר ולשם כך נשתמש בקבוצת סלקטורים, כשבין סלקטור לסלקטור הסימן פסיק.

לדוגמא אם נרצה שהכותרות h1 ו-h2 וגם הפסקאות p, יופיעו בצבע אדום ממורכזות, נוכל לתת את ההגדרה כך:

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

```
h2 {  
  text-align: center;  
  color: red;  
}
```

```
p {  
  text-align: center;  
  color: red;  
}
```

אך נשים לב שיש כאן כפילות משולשת בקוד, על מנת למזער את הקוד ולחסוך במשאבים, נגדיר את רשימת הסלקטורים שיקבלו את העיצוב הנ"ל, כשבין אחד לאחד הסימן פסיק כך:

```
h1, h2, p {  
  text-align: center;
```

```
color: red;
}
```

תרגיל מס 23

1. צרו מסמך html חדש.
2. העתיקו למסמך ה-html את הקוד הבא:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>העמוד שלנו</title>
  </head>
  <body dir="rtl" >
    <h1>זו כותרת ראשית</h1>
    <p>זו פסקה</p>
    <p>זו פסקה נוספת</p>
  </body>
</html>
```

3. צרו מסמך CSS חדש.
4. צרו קישור למסמך ה-CSS.
5. הגדירו שכל הפסקאות בדף יהיו בצבע אדום.

תרגיל מס 24

1. צרו מסמך html חדש.
2. העתיקו למסמך ה-html את הקוד הבא:

```
<!DOCTYPE html>
```

```
<html>

  <head>

    <meta charset="utf-8">
    <title>העמוד שלנו</title>
  </head>

  <body dir="rtl" >

    <h1>זו כותרת ראשית</h1>

    <p id="paragraph1">זו פסקה</p>

    <p>זו פסקה נוספת</p>

  </body>

</html>
```

1. צרו מסמך CSS חדש.
2. צרו קישור למסך ה-CSS.
3. הגדירו שצבע הפסקה עם ה-id paragraph1 יהיה אדום.

1. צור מסמך html חדש.

2. העתק למסמך ה-html את הקוד הבא:

```
<html>

  <head>

    <meta charset="utf-8">
    <title>העמוד שלנו</title>
  </head>
  <body>
    <h1>מוצרים</h1>
    <ul>
      <li class="odd">1 מוצר</li>
      <li class="even">2 מוצר</li>
      <li class="odd">3 מוצר</li>
      <li class="even">4 מוצר</li>
      <li class="odd">5 מוצר</li>
      <li class="even">6 מוצר</li>
    </ul>
  </body>
</html>
```

1. צור מסמך CSS חדש.

2. צור קישור למסך ה-CSS.

3. הגדר שצבע הפסקאות של המוצרים הזוגיים יהיה בצבע כחול (הפסקאות עם הקלאס even, צבע כחול - blue).

4. הגדר שצבע הפסקאות של המוצרים האי-זוגיים יהיה בצבע אדום (הפסקאות עם הקלאס odd, צבע אדום - red).

1. צור מסמך html חדש.

2. העתק למסמך ה-html את הקוד הבא:

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">
    <title>העמוד שלנו</title>
  </head>

  <body dir="rtl" >

    <h1>זו כותרת ראשית</h1>

    <h2>זו כותרת נוספת</h2>

    <p>זו פסקה</p>

    <p>זו פסקה נוספת</p>

    <h2>זו כותרת נוספת</h2>

    <p>זו פסקה</p>

    <p>זו פסקה נוספת</p>

  </body>

</html>
```

1. צור מסמך CSS חדש.

2. צור קישור למסך ה-CSS.

3. הגדר שהכותרות הראשיות וכותרות המשנה תצבענה בצבע כחול, (השתמש במינימום קוד).

סלקטור לאלמנטים מקוננים

לעיתים נרצה להגדיר הגדרת CSS לאלמנט מסוים שמקונן בתוך אלמנט אחר, לדוגמא קישור שממוקם בתוך פריט ברשימה, אולם לא נרצה שהגדרה זו תשפיע על שאר הקישורים במסמך, אלא רק על קישורים הממוקמים בתוך תגית `li`, קישורים שהם פריט ברשימה.

נוכל להגדיר הגדרת CSS לאלמנטים מקוננים בצורה כזו שנציג את שם האלמנט הראשי או ה-`id` שלו או ה-`class` שלו, לאחר מכן נשים ריווח ולאחר מכן את שם האלמנט הפנימי. (הקינון יכול להיות גם ביותר מרמה אחת).

לדוגמא:

```
ul li a { color: orange; }
```

בדוגמא זו הגדרנו לכל הקישורים שממוקמים כפריט ברשימה בלתי ממוספרת, `a` שבתוך `li` שבתוך `ul`, שיקבלו עיצוב בצבע כתום.

```
ul li.new a { color: orange; }
```

בדוגמא זו הגדרנו שכל הקישורים שממוקמים כפריט ברשימה בלתי ממוספרת, שלפריט, ל-`li` יש את הקלאס `new`, יקבלו עיצוב בצבע כתום. `a` שבתוך `li` שיש לו את הקלאס `new` שבתוך `ul`.

אם נרצה לבדוק שכתבנו נכון את הסלקטור, נקרא את הסלקטור בשילוב המילה **שבתוך** עבור כל ריווח בין חלקי הסלקטור, לדוגמא את הסלקטור הבא:

```
ul li a { color: orange; }
```

ננסה לקרוא כך: `a` שבתוך `li` שבתוך `ul`.

להגדרות צמודות, ללא ריווח כמו `li.new`, נוסיף את המילה **שהוא** לבדיקת והבנת הסלקטור. (אובייקט עם קלאס `new` שהוא `li`).

חוזק סלקטורים

כמה הגדרות להגדרת חוזק סלקטורים, על מנת לקבוע מי גובר בשעה שנתקלים בקונפליקט בין הגדרות CSS, על קונפליקט הנוצר כתוצאה מהגדרה בכמה מקומות, כלומר בקובץ ה-CSS החיצוני או בגוף התגית או בראש המסמך, הוסבר בתחילת השיעור.

במידה והקונפליקט יושב באותה רמה של קובץ, לדוגמא הקונפליקט נוצר בקובץ CSS מקושר נעזר במספר כללים להבנה מי יגבר מבין הסלקטורים המתנגשים:

1. מיקום ההגדרה במסמך ה-CSS, ההגדרה המאוחרת גוברת

לדוגמא אם הגדרנו במסך ה-CSS את שתי ההגדרות הבאות:

```
P { color: orange; }
```

```
P { color: green; }
```

ההגדרה השנייה, היא הגוברת והפסקאות תקבלנה צבע ירוק.

2. ספציפיות מקבלת יתרון

ככל שההגדרה ספציפית יותר, לדוגמא אם הגדרנו במסמך ה-CSS את ההגדרות הבאות:

```
ul li a {color: red; }
```

```
a {color: blue; }
```

למרות שהגדרת הצבע הכחול מאוחרת יותר, כל קישור שהוא בתוך רשימה בלתי ממוספרת יצבע בצבע אדום.

חישוב ספציפיות לסלקטור

ניתן לחשב את משקל הספציפיות ולתת ציון לסלקטור על מנת לבדוק את רמת הספציפיות שלו:

- כל תגית HTML, שם האלמנט בסלקטור שוקלת 1. לדוגמא **p**.
- כל קלאס או מאפיין (attribute) בסלקטור שוקל 10. לדוגמא: **[land=en]** או דוגמא נוספת **..new**.
- כל ID בסלקטור שוקל 100, לדוגמא **#contact**.

תרגיל 27 חשב את משקל הסלקטורים הבאים:

חשבו את משקל הסלקטורים הבאים, קבעו מי מבניהם גובר!

P _____

div p (div הוא שם של תגית) _____

#new p _____

.tree p _____

body #new .con p _____

מי הסלקטור הגובר?

צבעים ורקעים בCSS

צבעים ורקעים ב-CSS

Css מאפשר להגדיר צבעים וצבעי רקע לכל אלמנט בדף, ניתן לבחור מקשת רחבה מאד של צבעים, יש 256 בחזקת 3 (סה"כ 16,777,216) צבעים.

יש כמה דרכים לקבוע צבע וצבע רקע. כל ההגדרות הבאות מגדירות את הצבע האדום:

1. Red
2. Rgb(255,0,0)
3. Rgb(100%,0%,0%)
4. #ff0000
5. #f00

- ההגדרה הראשונה, הינה הגדרה מילולית המכנה את שם הצבע.

רשימה של חלק מהצבעים, תוכלו לצפות בשמותיהם:

https://www.w3schools.com/tags/ref_colornames.asp

- Rgb- ההגדרה השנייה מבוססת על ערבוב צבעים, כשהבסיס לערבוב הם צבעי היסוד

red green blue, הערך שניתן לתת לכל צבע הוא בין 0 ל-255, כלומר בדוגמא הערך 255 במאפיין הראשון, ו-0 בשאר המאפיינים, נותן תוצאה של צבע אדום מלא.

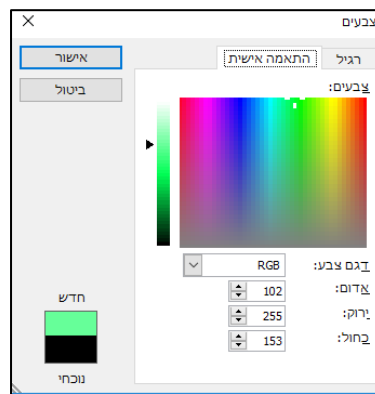
ניתן להגדיר גם שקיפות לצבע באמצעות הוספת ערך נוסף לשיטת להגדרת הצבע כך:

Rgb(255,0,0,0.5)

הערך שניתן לתת למאפיין הרביעי הוא ערך בין 0 ל-1, כשהערך 1 נותן אטימות מוחלטת והערך 0 נותן שקיפות מוחלטת, הערכים בניהם נותנים רמות שקיפות שונות.

- ההגדרה השלישית דומה בהגדרה לשנייה, רק שמגדירה באחוזים, כשהערך המקסימאלי הוא 100%, כך ש100% מהצבע red, כמובן נותן תוצאה של צבע אדום.
- ההגדרה הרביעית היא הגדרת הצבע בשיטת hexadecimal (שיטת ספירה בבסיס 16), בשיטה זו הצבעים מוגדרים על ידי צירוף שישה תווים המכילים ספרות בין 0 ל-9 ואותיות בין a ל-f. הצבע הלבן לדוגמה בשיטה זו יוצג כך: #ffffff. אם הצבע, או חלקו מוגדר מתווים זהים ניתן לצמצם את שם הצבע לשלשה תווים. כך: #fff, או כמו בדוגמאות 4 ו-5.

על מנת להגדיר את שם הצבע שברצונכם להשתמש בו, ניתן להגדיר את הצבע באחת מתוכנות ה-officce, ולהעתיק את הגדרת הצבע מהפרמטרים rgb, לדוגמא:



את הצבע הירוק שדגמנו בדוגם הצבעים בתוכנת הוורד נתאר ב-CSS כך:

`Rgb(102,255,153)`

ניתן גם להשתמש בשלל דוגמי צבעים הקיימים באינטרנט, תגלגלו את המילה: `color picker`

ותקבלו שלל תוצאות של דוגמי צבעים שתוכלו להעזר בהם, מספר דוגמאות שעלו בתוצאות ראשונות בגוגל:

- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Colors/Color_picker_tool
- https://www.w3schools.com/colors/colors_picker.asp

<https://htmlcolorcodes.com/color-picker/>

צבע וצבע רקע color ו- background-color

לכל אובייקט ניתן לקבוע צבע וצבע רקע, הצבע הוא הצבע שיוגדר לטקסטים וצבע רקע הוא צבע הרקע שיוצג ברקע האלמנט.

הגדרת צבע לאלמנט:

```
h1 { color: yellow; }
```

כך הגדרנו שצבע הכותרת הראשית יהיה צהוב.

על מנת להגדיר גם צבע רקע לכותרת נשתמש בהגדרה הבאה:

```
h1 {  
  
    color: yellow;  
  
    background-color: blue;  
  
}
```

הוספנו את ההגדרה background-color, שתגדיר שצבע הרקע של הכותרת יהיה כחול.

תרגיל 28

1. פתחו את הקובץ color.html.
2. פתחו את תכנה ה-notepad++.
3. צרו מסמך CSS חדש.
4. שמרו את המסמך בשם style.css.
5. הגדירו את צבע הטקסט לכל פסקה בהתאם לצבע שהיא מתארת.
6. קבעו צבע רקע לתגית body, בצבע בהיר, כך שיראו את כל הטקסטים במסמך.

background-image - תמונת רקע

ניתן להגדיר ב-CSS קובץ תמונה שיהווה כרקע של אלמנט מסוים, המאפיין מקבל כערך כתובת url המצביעה על מיקומו של הקובץ שיהווה כרקע.

לדוגמא:

```
body { background-image: url(img/flower.jpg); }
```

(הערה: כמובן הכתובת הנ"ל מתייחסת באופן יחסי לדף במקרה שהקובץ שמור בתיקייה img, אך אם קובץ התמונה ממוקם באותה תיקייה בה הדף ממוקם, כתובת ה-url תהיה רק שם הקובץ ללא הקידומת (img/

חזרת תמונת הרקע - background-repeat

ניתן להגדיר באיזה אופן תמונת הרקע תחזור על עצמה באמצעות המאפיין:

background-repeat, המאפיין יכול לקבל את אחד מארבעת הערכים הבאים:

1. **Repeat** - הרקע יחזור על עצמו בשני הצירים.

לדוגמא הקוד הבא:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <meta name="viewport" content="width=device-width,initial-  
cale=1.0">
```

```
  <title>כותרת עמוד</title>
```

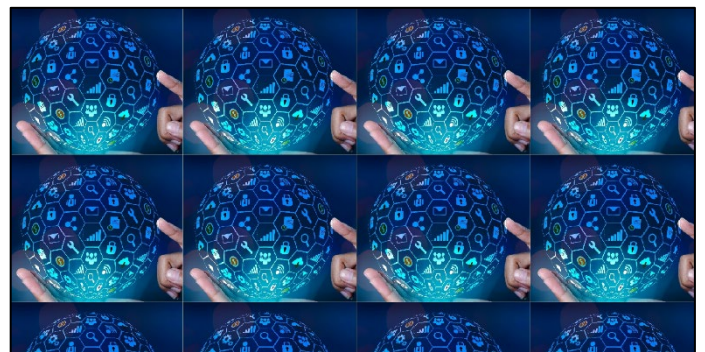
```
  <style type="text/css">
```

```
    body {
```

```
      background-image:
```

```
url(internet.jpg);
```

```
      background-repeat: repeat;
```



```
    }  
  </style>  
</head>  
  
</body>  
</html>
```

2. **no-repeat** - הרקע לא יחזור על עצמו

שנינו את קוד ה-CSS במסמך הקודם:

```
body {  
  background-image:  
url(internet.jpg);  
  background-repeat: no-repeat;  
}
```



3. **repeat-x** - הרקע יחזור על עצמו על ציר ה-x (הציר האופקי)

שנינו את קוד ה-CSS במסמך הקודם:

```
body {
```

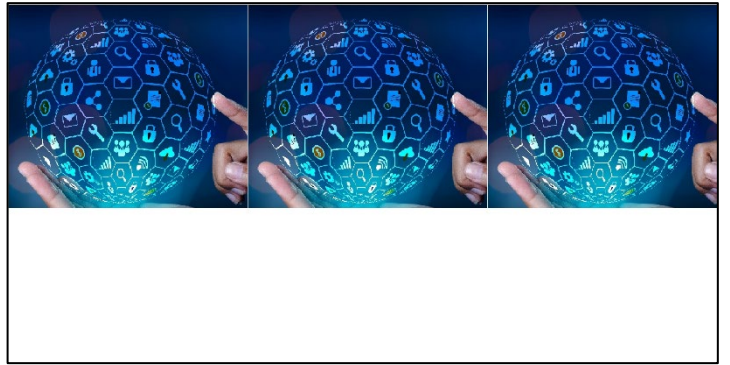
```

background-image:
url(internet.jpg);

background-repeat: repeat-x;

}

```



4. **repeat-y** - הרקע יחזור על עצמו על ציר ה-y (הציר האנכי)

שנינו את קוד ה-CSS במסמך הקודם:

```

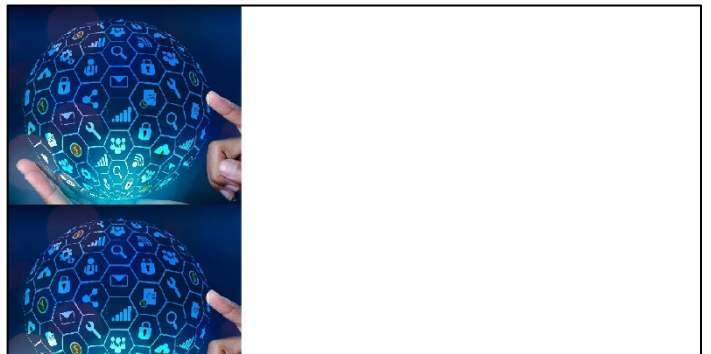
body {

background-image:
url(internet.jpg);

background-repeat: repeat-y;

}

```



background-position - מיקום הרקע

ברירת המחדל היא שהתמונה מתמקמת במיקום 0,0 ביחס לאלמנט לו היא משמשת כרקע, לעיתים נרצה לקבוע מיקום שונה על ציר ה-x או ה-y, נעשה זאת על ידי שימוש במאפיין background-position, מאפיין זה מקבל שני ערכים, הערך הראשון מייצג את המיקום על ציר ה-x ואילו הערך השני מציין את מיקום התמונה על ציר ה-y.

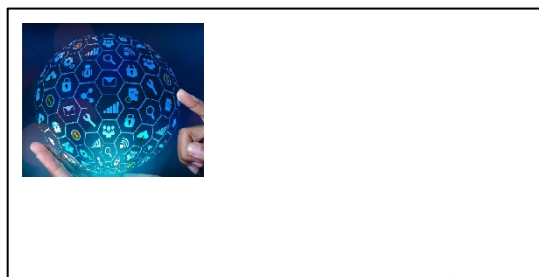
הערכים נמדדים מהנקודה העליונה השמאלית של האלמנט, נקודה זו היא נקודת ה-0,0, ערך חיובי יורד כלפי מטה בציר ה-y וזו ימינה בציר ה-x, ערכים שליליים יזיזו בכיוון הנגדי. לדוגמא:

```
background-position: 50px 50px;
```

ערך זה מגדיר שהתמונה תזוז ימינה 50 פיקסלים, ולמטה 50 פיקסלים.

כך שהקוד הבא, יראה כך:

```
body {  
    background-image:  
url(internet.jpg);  
    background-repeat: no-repeat;  
    background-position: 50px 50px; }  
}
```



נשים לב שהתמונה לא צמודה לחלק העליון של המסך, אלא ממוקמת במרחק של 50 פיקסלים מכל כיוון.

ניתן גם להגדיר למאפיין background-position את ערכי ה-x וה-y גם באחוזים,

לדוגמא:

```
body {  
    background-image:  
url(internet.jpg);  
    background-repeat: no-repeat;  
    background-position: 50% 0%; }
```



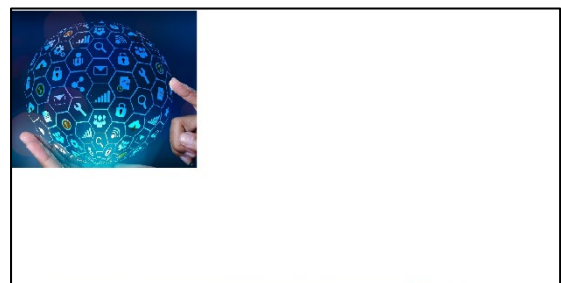
נשים לב, כיוון שקבענו בציר ה-x 50% תמונת הרקע התמקמה במרכז הציר האופקי.

ניתן גם להגדיר למאפיין background-position את אחד מתשעת הערכים המילוליים הבאים, כשכל אחד מהערכים מציין אחד מתשעה מיקומים האפשריים בחלוקת האובייקט לתשע כך:

left top	center top	right top
left center	center center	right center
left bottom	center bottom	right bottom

הערך left top יגדיר שתמונת הרקע תצמד לפינה השמאלית העליונה כך:

```
body {  
    background-image:  
url(internet.jpg);  
    background-repeat: no-repeat;  
    background-position: left top;  
}
```



background-attachment - חיבור הרקע

ניתן להגדיר את האופן בו הרקע יקובע או ינוע באמצעות השימוש במאפיין `background-attachment`, מאפיין זה מגדיר אם הרקע יתקבע למסך, כך שבמצב גלילה התמונה לא תזוז ושאר האלמנטים במסך יזוזו, המאפיין יקבל את אחד משני הערכים הבאים:

1. `Scroll` - הרקע יזוז עם האלמנט במצב גלילה.
2. `Fixed` - הרקע יתקבע למסך במצב גלילה, ולא ינוע.

לדוגמא:

```
body {  
  
    background-attachment: fixed;  
  
}
```

הגדרה זו קובעת שהרקע של ה-`body` לא יזוז, גם במצב של גלילה.

Background-size

המאפיין `Background-size` מגדיר את גודל תמונת הרקע, מאפיין זה מקבל את אחד מהערכים הבאים:

1. `Cover` - התמונה תמתח כך שתמלא את כל שטח המסך, באופן זה פעמים רבות אם התמונה לא בפרופורציה מדויקת לזו של המסך או לאובייקט אותו היא משמשת כרקע, התמונה תמשיך לגדול, בכדי לכסות את המסך או האובייקט בכל מחיר, וחלקים ממנה לא יוצגו כי לאחר שתמתח במימד אחד, אופקי או אנכי, יתכן שהמימד השני יהיה גדול מהאלמנט.
2. `Contain` - התמונה תמתח בשאיפה למלא את כל השטח, אך ברגע שתגיע למצב שאם תגדל עוד "תאבד" חלקים, תעצר מלגדול, תשמור על תצוגה שלמה של התמונה.
3. `רוחב וגובה` - המאפיין יכול לקבל גובל מסוים, באמצעות שני ערכים של רוחב וגובה הערך הראשון קובע את הרוחב, הערך השני קובע את הגובה. אם נכתוב רק ערך

אחד, השני מוגדר "אוטומטי". באותו גודל, ניתן לקבוע את הערכים בפיקסלים או באחוזים.

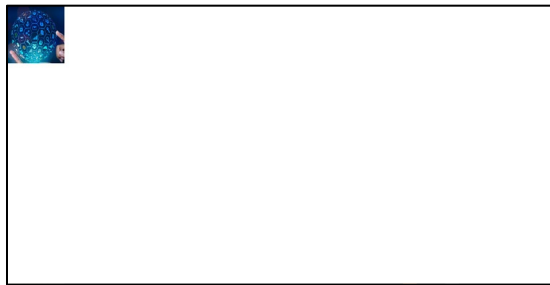
לדוגמא אם נתן את הערכים:

```
div {  
  background-image: url(internet.jpg);  
  background-repeat:no-repeat;  
  background-size:200px;  
}
```

או כשנתן את הערכים:

```
div {  
  background-image: url(internet.jpg);  
  background-repeat:no-repeat;  
  background-size:200px 200px;  
}
```

נקבל את אותה התוצאה:



שילוב המאפיינים במשפט מקוצר

ניתן לקצר ולשלב את המאפיינים על ידי שימוש במאפיין `background` בלבד והגדרת שאר המאפיינים לפי הסדר הבא:

```
background: red url(internet.jpg); no-repeat scroll 50px 50px;
```

יש לשמור על סדר ההגדרה: צבע רקע, מילוי תמונה, אופן החזרה, אופן חיבור הרקע, גודל הכיסוי, ניתן להגדיר רק חלק מהערכים, אין חובה להשתמש בכלם, במידה והתמונה לא

מתפרסת על כל הרקע והוגדר גם צבע רקע החללים שאינם מכוסים בתמונה יקבלו את הצבע המוגדר.

תרגיל 29

1. צרו מסמך html חדש
2. קבעו לתגית body רקע עם תמונת רקע
3. קבעו שתמונת הרקע תחזור על עצמה בציר האופקי ובציר האנכי
4. שלבו את כל ההגדרות למשפט מקוצר

תרגיל 30

1. פתחו את המסמך tar-1.html
2. קבעו לתגית body תמונת רקע
3. קבעו שתמונת הרקע תמתח ותמלא את כל שטח המסך
4. קבעו שתמונת הרקע תהיה מחוברת לרקע ולא תזוז בעת הגלילה.

ריבוי תמונת רקע

ניתן להגדיר לאלמנט אחד יותר מתמונת רקע אחת, כאשר נגדיר יותר מתמונת רקע אחת זו שמופיע קודם בפקודה תהיה בשכבה עליונה יותר.

לדוגמא:

```
#example
{
background-image: url:(image1.jpg), url:(image2.jpg);
background-position: right bottom, left top;
background-repeat: no-repeat, repeat;
}
```

הגדרנו כיסוי כפול לרקע, והגדרנו לכל כיסוי מיקום, וחזרתיות.

רקע מעבר צבע

מבנה פקודת רקע הבנוי מצבעי מעבר:

Background: linear-gradient(כיוון, 1 צבע, 2 צבע, 3 צבע...);

את כיוון המעבר אין חובה לציין, אם לא ציינו המעבר יתבצע מלמעלה למטה (כיוון ברירת המחדל).

ניתן לקבוע מעבר בין שני צבעים ויותר, ככל שנציין יותר צבעים המעבר יהיה צבעוני ומגוון.

הכיוונים האפשריים הם:

to right - משמאל לימין

to bottom - מעבר אלכסוני

ניתן לקבוע את שיפוע המעבר גם בערך מספרי בצירוף ההגדרה deg.

לדוגמא:

```
#grad {  
  background-image: linear-gradient(-90deg, red, yellow);  
}
```

תוצאת הגדרה זו תתקבל שיפוע אלכסוני בזווית של 90 מעלות כך:



אם נקבע זווית של 0 מעלות לא תהיה להגדרה משמעות ונקבל מעבר הדרגתי מלמעלה כלפי מטה.

תרגיל 31

1. פתח מסמך html חדש.
2. הקלד במסמך חמשה משפטים בתוך פסקאות.
3. הגדר לכל פסקה מעבר בצבעי מעבר שונים.
4. קבע לכל מעבר זווית שיפוע שונה.

מעבר עם שקיפות

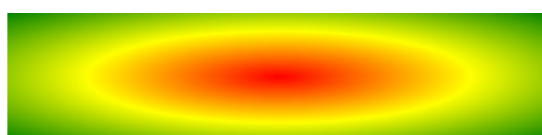
אם נכתוב את הצבע בשיטת ה-rgb (הוסבר בפרק צבעים ב-css) ונעביר פרמטר רביעי, נקבל מעבר עם שקיפות.

תרגיל 32

1. קבעו רקע מעבר צבעים המורכב מחמשה צבעים, לאחד הצבעים הגדר שקיפות קלה.
2. קבעו שהמעבר יהיה משמאל לימין.
3. שנו את כיוון המעבר, כך שהמעבר יהיה אלכסוני.

מעבר צבעים מעגלי

ניתן להגדיר שצורת מעבר הצבעים תהיה מעגלית בצורה כזו:



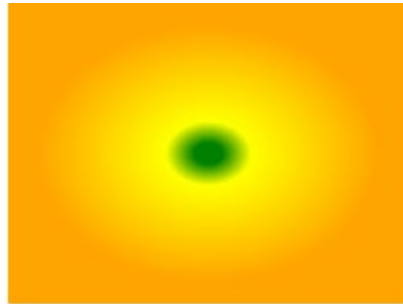
על ידי הגדרת המאפיין radial-gradient

```
#grad {  
  background-image: radial-gradient(red, yellow, green);  
}
```

ניתן להגדיר את רמת כמות הצבע לכל צבע במעבר על ידי הוספת ערך מספרי בציון הסימן אחוז לאחר כל צבע בהגדרה לדוגמא:

```
background-image: radial-gradient(green 5%, yellow 15%, orange 60%);
```

וכך זה יראה בתצוגה:



תרגיל 33

1. פתח מסמך html חדש
2. קבע רקע מעברי בסגנון מעגלי לתגית body.
3. הרקע יורכב מ-5 צבעים כל צבע יוגדר ל-20% מכלל המעבר.

כיוון מסמך, עיצוב טקסט ב CSS והטמעת פונטים של גוגל

Direction:rtl

תפקידו של תכונה זו להגדיר את כיוון המסמך, מאפיין זה מקבל את אחד משני הערכים הבאים: rtl (right to left מימין לשמאל) או ltr (left to right משמאל לימין).

כאשר נגדיר הגדרה זו לאלמנטים המכילים טקסטים, ההגדרה תשפיע על כיוון הכתיבה. בעברית כיוון הכתיבה הוא מימין לשמאל (rtl), עקב כך הטקסט יוצמד לשוליים הימניים, באנגלית כיוון הכתיבה הוא משמאל לימין (ltr), עקב כך הטקסט יוצמד לשוליים השמאליים.

מאפיין זה אינו משפיע רק על הצמדת הטקסט, אלא גם על זרימת כל האלמנטים במסמך.

תרגול 34

1. פתחו את המסמך Sentences.html
2. לאיזה כיוון צמודים הטקסטים במסמך?
3. צרו מסמך CSS חדש.
4. הגדירו באמצעות CSS את המאפיין direction לתגית-body, כך שהטקסט יוצג מימין לשמאל.
5. קשרו את מסמך ה-CSS לקובץ ה-HTML.
6. צפו בתוצאות.

עיצוב טקסטים

ניתן לקבוע את סוג הגופן גודל הגופן, מרווחים בין התווים ובין השורות, צבעים ועוד...

צבע טקסט

כפי שכבר ראינו צבע טקסט נקבע באמצעות המאפיין color

לדוגמא:

P{color: red}

הגדרה זו קובעת שכל הטקסטים בפסקאות יצבעו בצבע אדום.

סוג גופן - font-family

את סוגי הגופנים לטקסטים באתר מגדירים באמצעות CSS, על מנת שתצוגת הטקסט יראה על מחשב הגולש כפי שתכננו, יש צורך שהפונטים שהאתר משתמש בהם ישבו על מחשבו של הגולש

מסיבה זו נגדיר גופן נפוץ שקיים ברוב המחשבים, או לחילופין פונטים של גוגל המוטמעים ברשת הכללית, במידה ונרצה להשתמש בפונטים ייחודיים יותר, נוכל להטמיע אותם בתוך האתר.

בכל מקרה, גם אם הגדרנו פונטים מיוחדים יותר, נגדיר גם אפשרות של תצוגת פונטים פשוטים יותר, למקרה שהפונט שהגדרנו לא נמצא במחשבו של הגולש, או שהאתר לא מצליח לטעון את הגופן ממקורו.

פונטים בסיסיים שנמצאים כמעט בכל מחשב: Arial, Times New Roman, Tahoma

הגדרת סוג הגופן תתבצע באמצעות המאפיין font-family לדוגמא:

```
font-family: arial, serif, 'Time New Roman';
```

שם פונט המורכב מיותר ממילה אחת יש לשים אותו תחום בגרשיים או מרכאות.

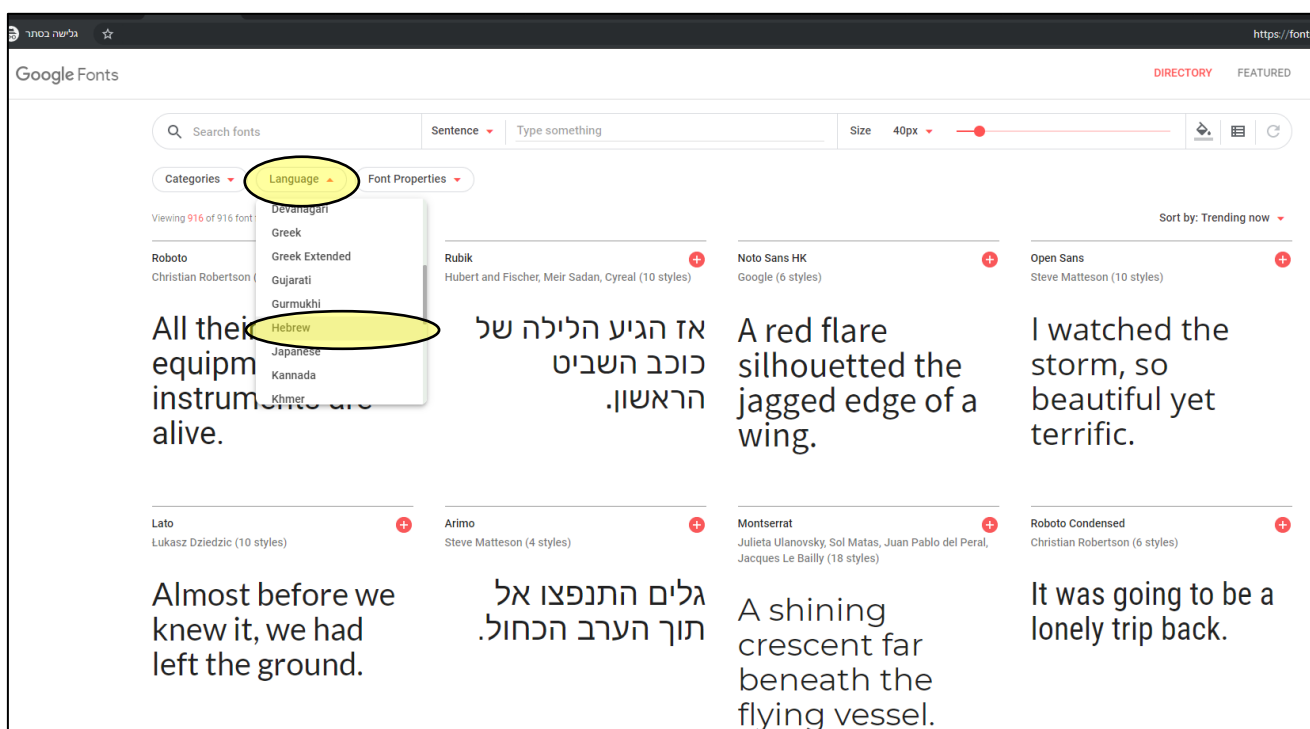
הטמעת פונטים של גוגל

ישנה גם אפשרות להטמיע פונטים של גוגל מתוך הרשת.

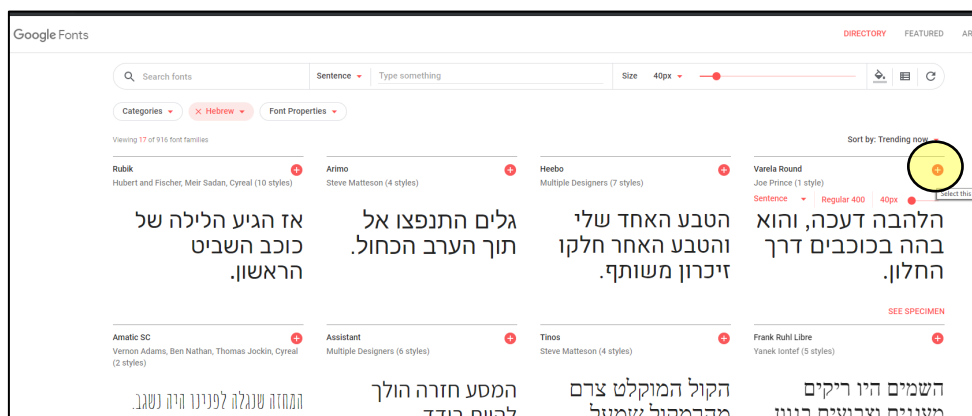
על מנת להטמיע פונט של גוגל דבר ראשון עלינו להיכנס לאתר של גוגל:

<https://fonts.google.com/>

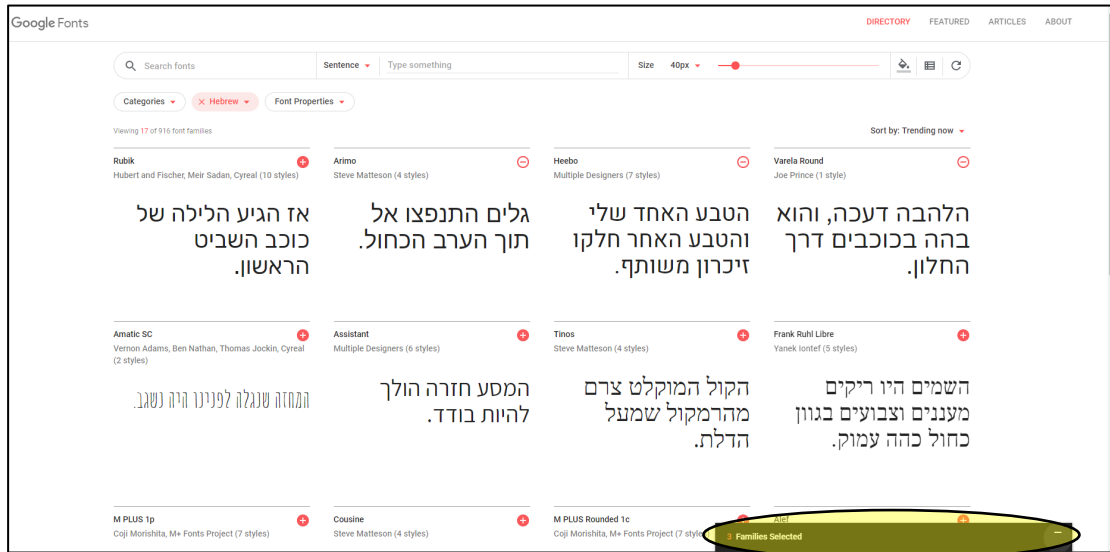
אם נרצה להשתמש בפונט בעברית נבחר בלחצן השפה, language בשפה העברית, Hebrew.



נלחץ על הלחצן + המופיע לצד הגופנים שברצוננו להטמיע באתר.



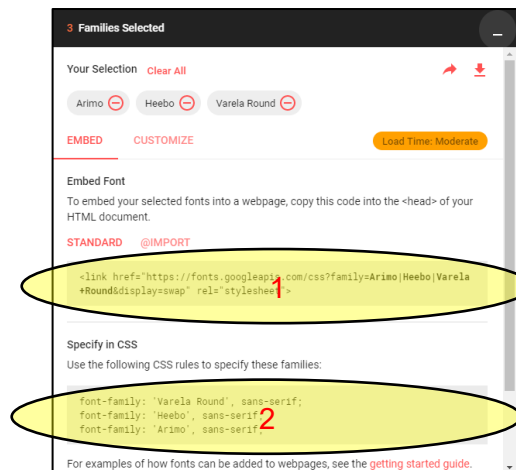
נקבל בתחתית חלון האתר שורה בו מצוין מספר הגופנים שבחרנו.



נלחץ על הסימון מינוס (-) המופיע לימין שורה זו ונקבל חלון בו שתי שורות שנעתיק אותן.

1. השורה link נעתיק אותה למסמך ה-`html`, נניח שורה זו בתגית ה-`head/`

2. מאפייני ה-`css`, תיאור שמות הגופנים בגיליון ה-`css`.



תרגיל 35

1. פתחו את המסמך `flowers.html`.

2. הטמיעו במסמך 2 גופנים עבריים של גוגל.

3. הגדר בעזרת CSS גופן אחד לכותרות וגופן נוסף לשאר הטקסטים - לפסקאות.

גודל גופן - font-size

באמצעות המאפיין font-size נקבע את גודל הגופן, מיותר לציין שקביעת גודל גופן גדול לא תחליף את הכנסת התוכן הראוי לכותרות בתגית כותרת מתאימה, הכנסת ערכים בתגית כותרת חשובה מאד מבחינת קידום האתר וקריאת מנועי החיפוש.
לדוגמא:

```
font-size: 18px;
```

משקל הגופן - font-weight

לקביעת משקל הגופן, נשתמש במאפיין font-weight, מאפיין זה מאפשר לעבות את הגופן הא קבל את אחד מהערכים הבאים:
bold - מודגש, להדגשת טקסטים.
normal - בלתי מודגש.

סגנון גופן - font-style

ניתן גם לקבוע את סגנון הגופן, אם הגופן יהיה נטוי, שימוש בערך italic או oblique יגדיר את הגופן לנטוי, הערך normal יגדיר כתב רגיל בלתי נטוי.

גובה שורה - line-height

מאפיין זה קובע את גובה שורת הטקסטים, מאפיין זה מאפשר לשלוט במרווח בין השורות, תכונה זו מקבלת ערכים מספריים כמו בגודל גופן, יתכן שנגדיר גודל גופן קטן וגובה שורה גדול יותר על מנת לקבל טקסט עם ריווח גדול בין השורות.
לדוגמא:

```
P {line-height: 200%;}
```

כך זה יראה בתצוגת הדפדפן:

החלבוב (שם מדעי: Euphorbia) הוא סוג צמחים ממשפחת החלבוביים. חלק ממיני החלבוב ושיחים רב-שנתיים. פרחי החלבוב ערוכים בתפרחות, הנראות כאילו יוצאות מתוך פרח אחד מתוכה אבקן אחד. במרכז הכוסית נמצאת שחלה בת שלוש מגורות הנישאת על עוקץ ארוך. רבים ממיני החלבוב משמשים כצמחי-נוי. בארץ ישראל גדלים בר כ-35 מינים שרובם עשבי-את התכשיר Euphorbium ששימש בעבר כסם הקאה וכסם משלשל, וכיום הוא מקובל בוטרי

מרווח בין התווים - letter-spacing

תכונה זו מגדירה את המרווח בין האותיות, הערך שנתן אחריה, יציין את המרחק הרצוי בין האותיות.
לדוגמא

```
P{ letter-spacing: 7px;}
```

ייצור פסקה מרווחת במרווח של 7 פיקסלים בין האותיות
כך זה יראה בתצוגת הדפדפן:

החלבוב (שם מדעי: Euphorbia) הוא סוג צמחים ממשפחת החלבוביים. חלק ממיני החלבוב הם עצים בשרניים המזכירים קקטוס, וחלקם עשבים חד-שנתיים ושיחים רב-שנתיים. פרחי החלבוב ערוכים בתפרחות, הנראות כאילו יוצאות מתוך פרח אחד בודד - כוסית. כל כוסית עטופה בעלים מאחים, ולכל פרח היוצא

ניתן גם להשתמש בערך שלילי, על מנת לצמצם את המרווח בין האותיות מהמרווח המקובל.
לדוגמא:

```
P{ letter-spacing: -2px;}
```

כך זה יראה בתצוגת הדפדפן:

החלבוב (שם מדעי: Euphorbia) הוא סוג צמחים ממשפחת החלבוביים. חלק ממיני החלבוב הם עצים בשרניים המזכירים קקטוס, וחלקם עשבים חד-שנתיים ושיחים רב-שנתיים. פרחי החלבוב ערוכים בתפרחות, הנראות כאילו יוצאות מתוך פרח אחד בודד - כוסית. כל כוסית עטופה בעלים מאחים, ולכל פרח היוצא מתוכה אבקן אחד. במרכז הכוסית נמצאת שחלה בת שלוש מגורות הנישאת על עוקץ ארוך. רבים ממיני החלבוב משמשים כצמחי-נוי. בארץ ישראל גדלים בר כ-35 מינים שרובם עשבי-את התכשיר Euphorbium ששימש בעבר כסם הקאה וכסם משלשל, וכיום הוא מקובל בוטרי. התנן E. antisiphilitica הנפוץ במקסיקו נכלל בעפם הנטוסיסם בשפת שפה (קטלניה) הנעמשת סוגר בחד בבפחות השגול וסוגר להב לזה לצביטה. בוקי החלבוב יש הלב שגפין רוח חוק, זו היא ההדמיה של החלבוב, החלבוב מנמא את האבילה שלו על ידי בעל חיים ביחז של החלב של.

נוכל להחזיר ערך להגדרה הסטנדרטית באמצעות המאפיין normal

לדוגמא:

```
p {letter-spacing: normal;}
```

מרווח בין המילים - word-spacing

תכונה זו מגדירה את המרווח בין המילים, הערך שנתן אחריה, יציין את המרחק הרצוי בין המילים.

לדוגמא

```
P{ word-spacing: 7px;}
```

ייצור מרווח של 7 פיקסלים בין המילים בפסקה.

כך זה יראה בתצוגת הדפדפן:

החלבוב (שם מדעי: Euphorbia) הוא סוג צמחים ממשפחת החלבוביים. חלק ממיני החלבוב הם עצים בשרניים המזכירים קקטוס, וחלקם עשבים חד-שנתיים ושיחים רב-שנתיים. פרחי החלבוב ערוכים בתפרחות, הנראות כאילו יוצאות מתוך פרח אחד בודד - כוסית. כל כוסית עטופה בעלים מאוחים, ולכל פרח היוצא מתוכה אבקן אחד. במרכז הכוסית נמצאת שחלה בת שלוש מגורות הנישאת על עוקץ ארוך.

שילוב המאפיינים במשפט אחד

ניתן לשלב את מאפייני הגדרת הטקסט במשפט אחד, חשוב מאד לשמור על סדר ההגדרות, ראשית יוגדרו המאפיינים `style, weight`, לאחר מכן `font-size` ו-`line-height` מופרדים בלוחסן (/) לבסוף נגדיר את רשימת סוגי הגופנים הרצויים, לא חייבים בכל הגדרת גופן להגדיר את כל הנתונים הללו, נתוני החובה הם רק `font-size` ו-`font-family`.

לדוגמא:

```
P{font: normal bold 14pt/16pt Arial,verdana,'sans serif'; }
```

תרגיל 36

1. פתחו את הקובץ `flowers.html`
2. קבעו שגודל הטקסט בפסקאות יהיה 18 פיקסלים.

3. קבעו שגובה השורה בפסקאות יהיה 200%.

4. קבעו מרווח בין המילים של 4 פיקסלים.

קישוט טקסט - **Text-decoration**

באמצעות המאפיין `text-decoration` נקבע אם אנו מעוניינים בקו תחתון לטקסט, קו עליון או קו על הטקסט (כעין מחיקה).

מאפיין זה מקבל את אחד מהערכים הבאים:

Overline - קו עליון מעל הטקסט, מלמעלה.

line-through - קו חוצה ממש על הטקסט כך: `line`, כעין מחיקה.

underline - קו תחתי.

underline overline - קו עליון ותחתון (כעין שורה לטקסט).

none - ללא קו תחתי, מאפיין זה שימושי פעמים רבות לביטול הקו התחתי המגיע עם קישורים, לא מומלץ לתת לטקסטים סתמיים קו תחתון, לעיתים רבות דבר מטעה ומבלבל עם קישורים.

text-decoration-color

מאפיין זה קובע את צבע הקו, לדוגמא:

```
a{text-decoration-color: red};
```

text-decoration-style

מאפיין זה קובע את סגנון הקו, קו רגיל, קו עבה, קו כפול וכדו'.

מאפיין זה מקבל את אחד מהערכים הבאים:

solid - להגדרת קו פשוט.

wavy - להגדרת קו גלי.

double - להגדרת קו כפול.

dashed - להגדרת קו מקווקו.

dotted - להגדרת קו מנוקד.

`P { text-decoration: underline; text-decoration-style: dashed; }`

כך זה יראה בדפדפן:

חלבולב (שם מדעי: Euphorbia) הוא סוג צמחים ממשפחת החלבולבנים. חלק ממיני החלבולבנים הם צמחים חד-שנתיים ושנתיים רב-שנתיים. פרחי החלבולב צהובים בתפרחות הגראות באזור עטופה בעלים מאוחים ולבל פרח היוצא מתוכה אבקן אחד במרכז הכנסות. גמצאת שח

הגדרת תכונה זו ללא הגדרת התכונה `text-decoration` תהיה חסרת משמעות.

שילוב המאפיינים במשפט אחד

ניתן לצרף את שלושת הפקודות יחדיו כך:

`p { text-decoration: underline red double; }`

בהגדרה זו הגדרנו קו תחתני בצבע אדום, מסגנון קו כפול.

תרגיל 37

1. פתחו מסמך html חדש
2. הוסיפו במסמך כותרת, כתוב בה "קישורים ללא קו"
3. הוסיפו במסמך 3 קישורים לאתרים שונים, בין תגיות הקישור יוצג שם האתר אליו הקישור מפנה.
4. הגדירו שהקו התחתני שתחת הקישורים לא יופיע.
5. הגדירו לכותרת קו תחתני גלי בצבע ירוק.

יישור טקסט - Text-align

התכונה `text-align` אחראית על יישור הטקסט בפסקה תכונה זו מקבלת את אחד מהמאפיינים הבאים:

left - יישור הטקסט לשמאל.

right - יישור הטקסט לימין.

center - יישור הטקסט למרכז.

justify - יישור הטקסט לשני הכיוונים.

לדוגמא:

```
P{ text-align:justify}
```

הגדרה זו תגדיר לכל הפסקאות להיות מיושרות משני הכיוונים.

transform-Text

תכונה זו מגדירה אם אותיות בשפה האנגלית תהינה באותיות גדולות, באותיות קטנות

תכונה זו מקבלת את אחד מהערכים הבאים:

Capitalize - ערך זה הופך את כל האותיות הראשונות בכל מילה לאותיות גדולות.

Uppercase - ערך זה הופך את כל האותיות לאותיות גדולות.

Lowercase - ערך זה הופך את כל האותיות לאותיות קטנות.

none - ערך הוא ערך ברירת המחדל, ומגדיר שהטקסט יהיה ללא כפי שהוא בא בברירת

המחדל, ללא שינוי של אותיות גדולות וקטנות.

הזחת טקסט - text-indent

תכונה זו מגדירה הזחה לטקסט, כביכול דוחפת את השורה מעט פנימה. תכונה זו מקבלת

ערך מספרי בפיסקלים או באחוזים או ביחסיות לגודל הגופן (em).

לדוגמא:

```
p { text-indent: 3em;}
```

כך זה יראה בתצוגה:

החלבוב (שם מדעי: Euphorbia) הוא סוג צמחים ממשפחת החלבוביים. החלבובים עשבים חד-שנתיים ושיחים רב-שנתיים. פרחי החלבוב ערוכים בתפרחות, הנכנסות עטופה בעלים מאוחים, ולכל פרח היוצא מתוכה אבקן אחד. במרכז הכוסית רבים ממיני החלבוב משמשים כצמחי-נוי. בארץ ישראל גדלים בר כ-35 מיני החלבוב. הנפוץ במרוקו מפיקים את התכשיר Euphorbium ששימש בעבר כסם הברזל. E. antisiphilitica הנפוץ במקסיקו מכיל ענפים המכוסים בשכבת שעווה (קנדליליה) לכה לצביעה. בנוסף בחלבוב יש חלב שמפיץ ריח חזק. זו היא ההתאמה של החיים בריח של החלב שלו.

תרגול לעיצוב טקסט - תרגיל 38

1. פתחו את הקובץ flowers.html
2. הגדירו לכל הפסקאות הזחה של 2em.
3. קבעו שכל הפסקאות תהינה מיושרות לשני הכיוונים.
4. שמרו את המסמך, וצפו בתוצאות.

עיצוב רשימות

עיצוב רשימות לא מסודרות

ניתן לשלוט בעיצוב הרשימה באמצעות מספר תכונות CSS:

list-style-type

תכונה זו מגדירה את סגנון הצורה המופיע לצד כל פריט ברשימה, הערכים שתכונה זו מקבלת הם:

circle - להגדרת עיגול ללא מילוי.

disc - להגדרת עיגול עם מילוי.

square - להגדרת ריבוע.

None - להסרת התבליט מהרשימה.

אפשרות נוספת להגדרת התבליט לצד כל פריט ברשימה היא תבליט תמונה, להגדרת תבליט תמונה יש צורך לשתול תמונה קטנה כתחליף לצורות הגאומטריות ברשימה, להגדרת תמונה במקום התבליטים יש צורך להשתמש בתכונה `list-style-image` וכערך נתן לתכונה זו את הערך `url()` כשבתוך הגרשיים נגדיר את שמו של הקובץ המיועד לשמש כתבליט, שם הקובץ ייתחם בגרשיים.

עיצוב רשימות מסודרות

`list-style-type`

בדומה להגדרת תבליט ברשימה לא ממוספרת נגדיר את צורת המספור ברשימה ממוספרת, ניתן להגדיר לתכונה זו מספר ערכים:

`upper-roman` - לספרות רומיות באותיות גדולות.

`lower-roman` - לספרות רומיות באותיות קטנות.

`lower-latin` - לאותיות קטנות באנגלית.

`upper-latin` - לאותיות גדולות באנגלית.

`decimal` - למספרים.

`Hebrew` - לאותיות בעברית.

`None` - להסרת המספור.

תרגול 39

1. צרו מסמך `html` חדש.
2. קבעו שהטקסט יהיה מכוון מימין לשמאל.
3. צרו רשימה ממוספרת של 10 בעלי חיים שונים.
4. הגדירו שהמספור יהיה באמצעות אותיות בעברית.
5. שמרו את הקובץ וצפו בתוצאה.

מודל הקופסא, גבולות ומרווחים

borders - גבולות

ניתן לקבוע לכל אלמנט גבולות ויזואליים, על מנת שנראה תצוגת הגבולות עלינו להגדיר עובי גבול, סגנון גבול וצבע גבול.

border-style - סגנון גבול

מאפיין זה קובע את סגנון הגבול, ניתן לקבוע לגבול מגוון של סגנונות:

solid - קו סולידי, קו פשוט.

dotted - קו מנוקד, קו המורכב מנקודות נקודות.

dashed - קו מקווקו, קו המורכב מקווקוים.

double - קו כפול.

groove - קו תלת מימדי.

סגנונות נוספים דומים לסגנון התלת מימדי: `outset`, `inset`, `ridge`.

Border-whidth - עובי גבול

באמצעות מאפיין זה נקבע את עובי הגבול, עובי הגבול יקבע בד"כ בערך פיקסלי כך:

```
div{ border-whidth: 2px; }
```

Border-color - צבע גבול

באמצעות מאפיין זה נקבע את צבע הגבול, הערכים שמאפיין זה מקבל, הם כל הצבעים האפשריים ב-CSS.

ניתן להגדיר את עובי הגבול, סוג הגבול, וצבע הגבול בפקודה אחת בקיצור, כך:

`border: 2px solid red;`

הגדרנו בפקודה זו גבול בעובי של 2 פיקסלים, מסגרון קו פשוט בצבע אדום.

שילוב מספר גבולות לאלמנט

יתכן שנרצה לשלב לכל כיוון גבול בעל אופי מעט שונה, או שנרצה גבול חלקי, כמו רק גבול עליון/תחתון/ימני/שמאלי, נוכל לעשות זאת על ידי הגדרות מסגרת לכיוון הרצוי:

border-top-width - רוחב גבול לגבול העליון.

border-right-width - רוחב לגבול ימני.

border-bottom-width - רוחב לגבול תחתון.

border-left-width - רוחב לגבול שמאלי.

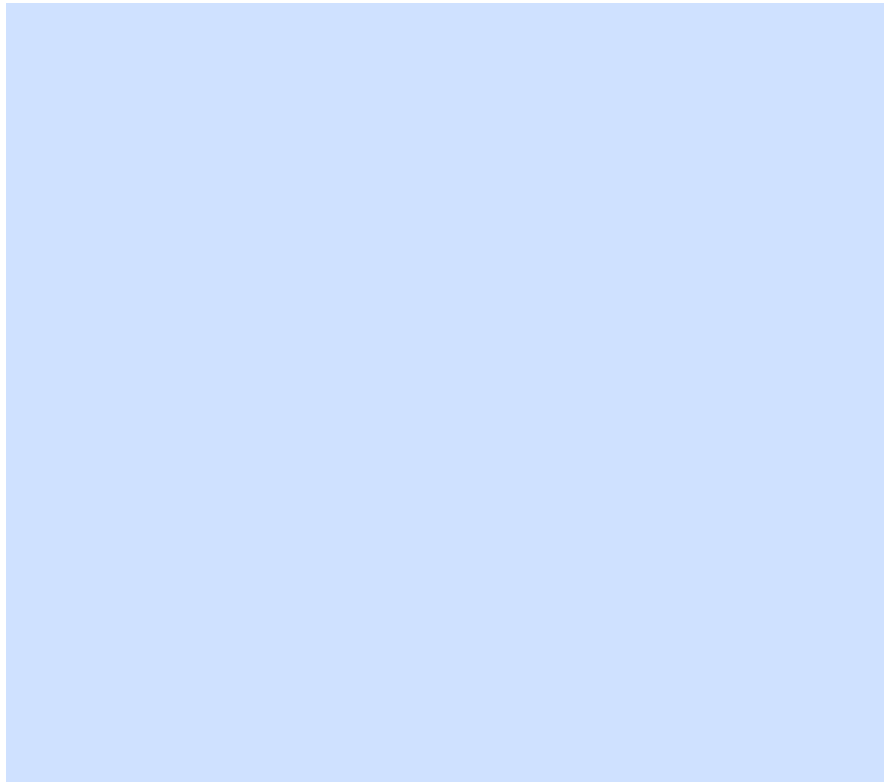
באותה הצורה שהגדרנו רוחב ספציפי לכל כיוון נוכל להגדיר את שאר המאפיינים, לדוגמא המאפיין צבע לגבול העליון נכתוב כך: `border-top-color`, או לסגרון קו שונה לגבול העליון נשתמש במאפיין `border-style-width`, על זו הדרך גם בכיוונים הנוספים.

תרגול 40

1. צרו מסמך html חדש.
2. הוסיפו למסמך 5 פסקאות ותמונה.
3. **לפסקה הראשונה** קבעו מסגרת עם גבול כתום.
4. **לפסקה השנייה** קבעו מסגרת עם גבול, בעובי של 5 פיקסלים, מסגרון מקווקו, בצבע שחור.
5. לשאר הפסקאות קבעו מסגרת עם גבול בעובי 3 פיקסלים, סגרון קו סולידי בצבע ירוק.
6. **לתמונה** קבעו מסגרת עם גבול משולב, גבול עליון בצבע כתום, גבול תחתון בצבע כחול, גבול ימני בצבע סגול, גבול שמאלי בצבע ירוק, שאר מאפייני הגבול קבעו כרצונכם.

מודל הקופסא - מרווחים בין אלמנטים

ניתן לקבוע מרווחים בין האלמנטים השונים במסמך ה-html, ניתן להגדיר ריווחים מארבעה כיוונים, עליון תחתון, ימני ושמאלי, ניתן להגדיר מרווח פנימי - מרווח בין תוכן האלמנט - תמונה, טקסט וכדו' למסגרת החיצונית של האובייקט (גם במידה ולא קיימת מסגרת נראית) אופן הריווחים בין האלמנטים מכונה בשם מודל הקופסא, כפי שנראה באיור הבא:



הריבוע התכול מציין את תוכן האלמנט, הריבוע הירוק מבטא את האזור שסביב התוכן ניתן להגדיר ריווח בין התוכן למסגרת המקיפה אותו, לדוגמא ריווח בין הטקסט למסגרת, ניתן להגדיר את הריווח גם במידה ואין מסגרת ממשית, הריבוע הצהבהב מבטא את המסגרת, הריבוע החיצוני הקרם מעליו מבטא את המרווח מהמסגרת החוצה, המרווח מהמסגרת לשאר האלמנטים, המרווח החיצוני של האלמנט.

את המרווחים נגדיר באמצעות המאפיינים הבאים:

margin - מרווח חיצוני

padding - מרווח פנימי

מאפיינים אלה יקבלו ערך מספרי, הערך יכול להיות ערך פיקסלי או ערך המצוין באחוזים. ניתן להגדיר לריווחים החיצוניים והפנימיים הגדרות מגוונות לכל כיוון הגדרה משלו באמצעות התכונות:

margin-left - מרווח חיצוני שמאלי.

margin-bottom - מרווח חיצוני תחתון.

margin-right - מרווח חיצוני ימני.

margin-top - מרווח חיצוני עליון.

לריווחים פנימיים:

Padding-left - מרווח פנימי שמאלי.

Padding -bottom - מרווח פנימי תחתון.

Padding -right - מרווח פנימי ימני.

Padding -top - מרווח פנימי עליון.

כתיבת כל ערכי הריווח בפקודה אחת

ניתן לכתוב את כל הערכים בפקודה אחת כשהערכים נעים עם כיוון השעון, הערך הראשון מתייחס לריווח העליון, הערך השני מתייחס לריווח הימני, הערך השלישי מתייחס לריווח התחתון והערך הרביעי מתייחס לריווח השמאלי.

הפקודה תראה כך:

Margin: 20px 25px 30px 35px;

אם קיימים שני ערכים זהים בכיוון נגדי (למעלה ולמטה, ימין ושמאל) נוכל לציין זאת כך:

margin: 10px 20px;

בפקודה זו קבענו ריווח חיצוני עליון ותחתון של 10 פיקסלים לכל כיוון, ריווח חיצוני ימני ושמאלי של 20 פיקסלים.

אם כל הערכים בעלי אותו ערך ניתן להגדיר את הפקודה באמצעות ערך מספרי בודד כך:

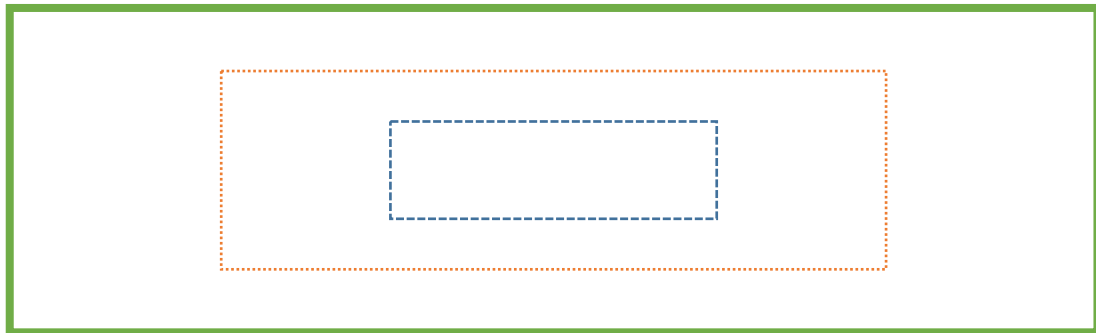
Margin: 0;

במידה והשתמשנו בערך אפס אין צורך לציין את שיטת המדידה.
קיימת אפשרות לתת ריווח חיצוני אוטומטי, מאפיין זה יגרום למרכז אלמנט ביחס לאלמנט המכיל אותו הפקודה תראה כך:

margin: auto;

תרגיל 41

נסו ליצור את מראה דומה למראה הבא באמצעות שילוב html עם הגדרות גבול ומרווחים ב-CSS.

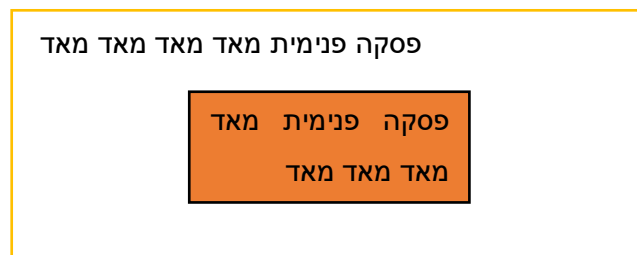


תרגיל 42

1. צרו קובץ html,
2. הוסיפו בתוכו שני משפטים
3. הקלידו בכל פסקה משפט או שתיים.
4. קבעו מרווח עליון ותחתון של 20 פיקסלים, ומרווח ימני ושמאלי של 50 פיקסלים.

תרגיל 43

צרו את המראה הבא באמצעות שילוב html ו-css.



1. צרו מסמך html חדש
2. הוסיפו 5 תמונות, קבעו מרווח בין התמונות.
3. הוסיפו 5 קישורים לאתרים שונים
4. קבעו צבע רק לאלמנט הקישור
5. קבעו מסגרת לאלמנט הקישור
6. קבעו מרווח בין הקישורים

border-radius - עיגול קצוות למסגרת

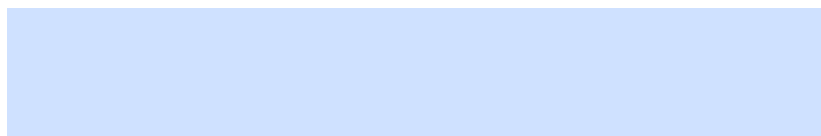
ניתן לעגל חלק מקצוות המסגרת או את כולה על מנת לעגל את קצוות המסגרת נשתמש במאפיין border-radius, המאפיין border-radius יכול לקבל 4 מאפיינים על מנת להגדיר לכל כיוון במסגרת עיגול עם מידת רדיוס שונה.

המאפיינים מציינים את כיווני המסגרת, כשהכיוונים נעים עם כיוון השעון: פינה עליונה שמאלית, פינה עליונה ימנית, פינה תחתונה ימנית פינה תחתונה שמאלית.

לדוגמא:

```
#example {
border: 2px solid red;
border-radius: 30px 15px 0px 0px;
}
```

מאפיין זה יצור את המראה הבא:



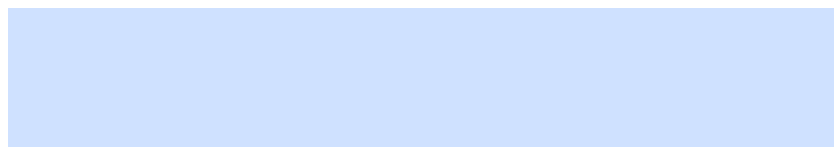
המסגרת קבלה קצוות מעוגלים בשתי הפינות העליונות, ואילו שתי הפינות התחתונות הן ללא הגדרת עיגול פינות.

במידה וכמה מהפינות מקבלות את אותן הזוויות ניתן להגדיר את הגדרה קצרה יותר ולצמצם בערכים.

לדוגמא:

```
#example2 {  
  
border: 2px solid red;  
  
border-radius: 30px 0px;  
  
}
```

מאפיין זה יצור את המראה הבא:



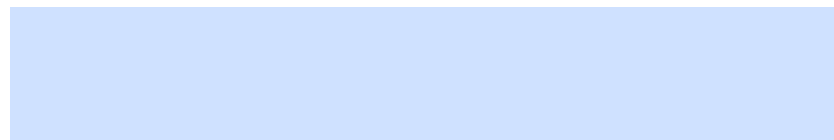
קצה ימני עליון וקצה שמאלי תחתון - זהים, בהגדרת עיגול ברמה של 30 פיקסלים.

קצה שמאלי עליון וקצה ימני תחתון - זהים, בהגדרת עיגול ברמה של 0 פיקסלים.

אם נעביר פרמטר בודד נקבל את כל קצוות המסגרת מעוגלים לדוגמא:

```
#example2 {  
  
border: 2px solid red;  
  
border-radius: 30px;  
  
}
```

מאפיין זה יצור את המראה הבא:



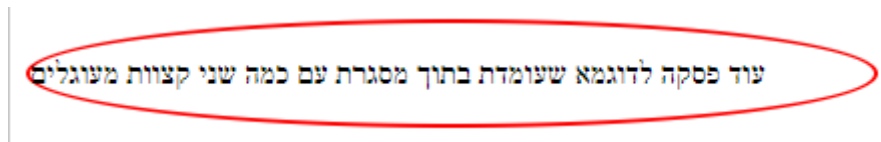
כל הקצוות קבלו הגדרת עיגול ברמה של 30 פיקסלים.

ניתן לקבוע את גודל זווית העיגול גם באחוזים.

לדוגמא:

```
#example2 {
```

```
border: 2px solid red;
border-radius: 50%;
}
```

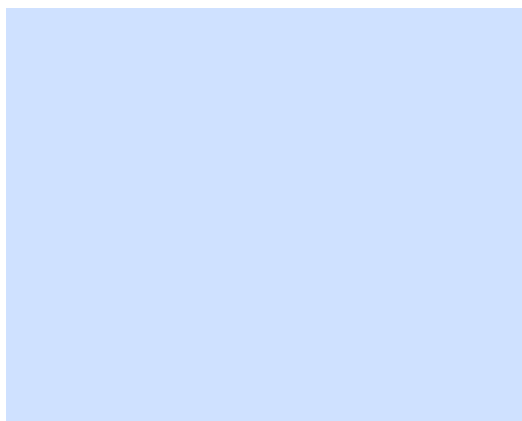


במאפיין זה הגדרנו עיגול קצוות מכל כיוון ברמה של 50%.

הערה: הגדרת מאפיין זה חייבת לבוא בצירוף הפקודה `border`, על מנת שהערכים יוחלו על מסגרת ולא על משהו בלתי קיים, במידה ולא הגדרנו את הפקודה `border` לא תהיה לפקודה משמעות כלל.

תרגיל 45

צרו את המראה הבא בעזרת שימוש ב- `html` ו- `css`, לא חשוב הצבע והזווית המדוייקת:

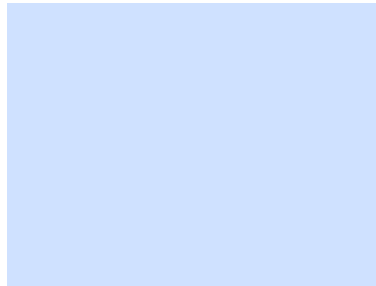


תשובה:

```
div {
border-radius: 25px;
background: #73AD21;
width: 200px;
height: 150px;
}
```


תרגיל 46 - אתגר

צרו את המראה הבא בעזרת שימוש ב- html ו-css, לא חשוב הצבע והזווית המדוייקת:

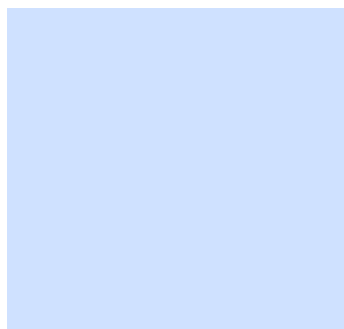


תשובה:

```
#rcorners {  
  border-radius: 0px 150px;  
  background: #73AD21;  
  width: 200px;  
  height: 150px;  
}
```

תרגיל 47

צרו את המראה הבא בעזרת שימוש ב- html ו-css, לא חשוב הצבע והזווית המדוייקת:



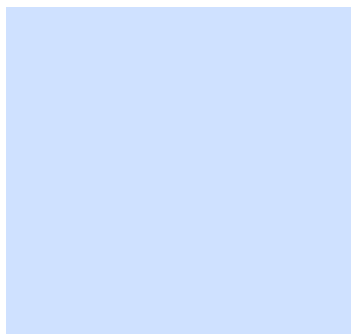
תשובה:

```
div {  
  border-radius: 50%;  
  background: #73AD21;  
  width: 150px;
```

```
height: 150px;
```

תרגיל 48

צרו את המראה הבא בעזרת שימוש ב- html ו-css, לא חשוב הצבע והזווית המדוייקת:

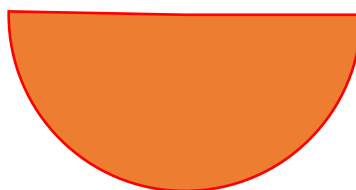


תשובה

```
div {  
  border-radius: 30% 0;  
  background: #73AD21;  
  width: 150px;  
  height: 150px;  
}
```

תרגיל 49

צרו את המראה הבא בעזרת שימוש ב- html ו-css, לא חשוב הצבע והזווית המדוייקת:



תשובה:

```
Div{  
  Border: 2px solid red;  
  background: orange;  
  width: 200px;
```

```
height: 100px;
```

```
border-radius: 0px 0px 200px 200px;}
```

טבלאות

טבלאות בhtml

בדף ה-html ניתן לבנות טבלאות, חשוב לציין שטבלאות נועדו להצגת תוכן במבנה טבלאי, אך לא לעימוד דפי ה-html, חלוקה לעמודות תבצע באופן שונה.

להגדרת טבלה בדף html נשתמש בתגית <table>, להגדרת שורה בטבלה נשתמש בתגית <tr>, להגדרת תא בשורה נשתמש בתגית <td>.

הגדרה בסיסית של טבלה:

```
<table>
```

```
<tr>
```

```
<td>שורה ראשונה, תא ראשון</td>
```

```
<td>שורה ראשונה, תא שני</td>
```

```
<td>שורה ראשונה, תא שלישי</td>
```

```
</tr>
```

```
<tr>
```

```
<td>שורה שניה, תא ראשון</td>
```

```
<td>שורה שניה, תא שני</td>
```

```
<td>שורה שניה, תא שלישי</td>
```

```
</tr>
```

תצוגת הקוד בדפדפן

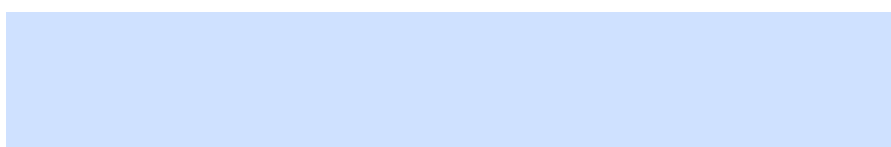
```
</table>
```

הצגת קווי מתאר לטבלה

טבלה שהוספנו במבנה ה-html תוצג בדפדפן ללא קווי מתאר אלא אם כן הגדרנו אותם. בכדי להציג את קווי המתאר של הטבלה, נגדיר בהגדרות ה-css את התכונה border לכל חלקי הטבלה, השורות העמודות והכותרות. לדוגמא:

```
td,tr,th,table{border: 2px solid black;}
```

לאחר הגדרת קווי המתאר, הטבלה תראה כך:



תרגיל 50

1. צרו מסמך html חדש.
2. בנו את הטבלה הבאה באמצעות קוד html, בשילוב הגדרות css במידת הצורך.

אנשי קשר			
שם	קומה	חדר	תפקיד
אבי ירב	כניסה	מספר 2	מנהל השבועות
מור מועלם	כניסה	מספר 8	מעצבת פנים
הביב לוי	שנייה	מספר 5	מהנדס חשמל

3. שמרו את מסמך ה-html, צפו בדפדפן שהמסמך נראה כראוי.

ניתן להוסיף כותרות לעמודות הטבלה, על ידי שימוש בתגית <th>, סגנון הטקסט של תאים אלו, הוא טקסט ממורכז ומודגש מעט.

לדוגמא:

```
<table border="1">
<tr>
<th>כותרת עמודה ראשונה</th>
<th>כותרת עמודה שניה</th>
<th>כותרת עמודה שלישית</th>
</tr>
<tr>
<td>שורה ראשונה, תא ראשון</td>
<td>שורה ראשונה, תא שני</td>
<td>שורה ראשונה, תא שלישי</td>
</tr>
<tr>
<td>שורה שניה, תא ראשון</td>
<td>שורה שניה, תא שני</td>
<td>שורה שניה, תא שלישי</td>
</tr>
```

</tr>

תצוגת הקוד בדפדפן

</table>

תרגיל 51

1. צרו מסמך html חדש.
2. בנו את טבלה באמצעות html ו-css במידת הצורך, עבור התצוגה הבאה:

שם	כתובת מרפאה	ימי עבודה
אנה שיין	הצופרים 52, תל אביב	ראשון ושלישי
יואב משיב	הדהלילים 38, ראשון לציון	שני ושלישי
עדן אבידן	המעופפים 85, גבעת היעלים	רביעי וחמישי

3. הגדירו את תאי השורה העליונה בתוך התגיות <th>.
4. שמרו את מסמך ה-html, צפו בדפדפן שהמסמך נראה כראוי.

rowspan מיזוג מספר תאים בשורה לתא אחד

ניתן למזג, לאחד מספר תאים סמוכים בשורה לתא אחד, בכדי לאחד מספר תאים לתא אחד, נוסף את המאפיין colspan לתגית אותה נרצה לפרוס על מספר תאים, נגדיר למאפיין colspan ערך מספרי להגדרת מספר התאים לאיחוד.

לדוגמא:

```
<table border="1">
```

```
<tr>
```

```
<th>שם</th>
```

```

    <th colspan="2">טלפון</th>
</tr>
<tr>
    <td>עמיר שמחון</td>
    <td>1234567891</td>
    <td>234567890</td>
</tr>
</table>

```

תצוגת הקוד בדפדפן

rowspan מיזוג מספר תאים בעמודה לתא אחד

ניתן גם למזג, לאחד מספר תאים סמוכים מאותה העמודה, באמצעות המאפיין rowspan, נגדיר למאפיין rowspan ערך מספרי להגדרת מספר התאים לאיחוד.

לדוגמא:

```

<table border="1">
<tr>
    <th>שם</th>
    <td>עמיר שמחון</td>
</tr>
<tr>
    <th rowspan="2">Telephone:</th>
    <td>1234567891</td>
</tr>
<tr>
    <td>234567890</td>

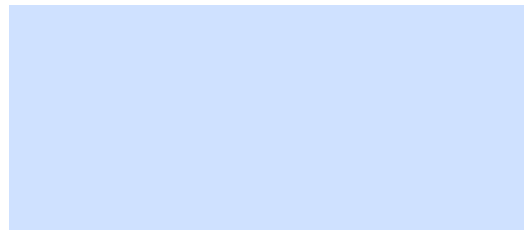
```

תצוגת הקוד בדפדפן


```
</tr>  
</table>
```

התכונה border-collapse

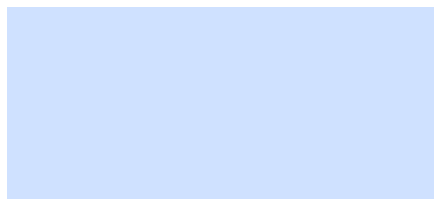
תצוגת טבלה שהגדרנו לה בהגדרות ה-css מסגור לחלקי הטבלה מתקבלת כמסגור כפול לכל תא ותא בטבלה כך:



על מנת לקבל תצוגה של רשת טבלאית ללא קווים כפולים נשתמש בתכונה border-collapse, תכונה זו מגדירה את תצוגת קווי המתאר בטבלה ומקבלת את אחד מהערכים הבאים:

Separate - ערך זה מגדיר גבולות מופרדים, כל תא יציג גבולות משלו, כתוצאה מערך זה נקבל טבלה עם קווי מתאר המוצגים כקו כפול, זוהי ברירת המחדל של תצוגת קווי מתאר בטבלה.

Collapse - ערך זה מגדיר איחוד של גבולות לגבול יחיד, ערך זה מכווץ את צורת תצוגת קווי המתאר בטבלה לקו בודד, לאחר הגדרת ערך זה תראה תצוגת הטבלה כך:





6. השתמשו במאפייני CSS להגדרת תכונת צבעי רקע, צבעי גופן וכיוון טקסט.

7. הגדירו את רוחב הטבלה ל-100%

(הגדרת רוחב הטבלה תתבצע כך: `table { width:100%;}`).

8. הגדירו מרכז לטקסטים.

9. הגדירו שגבולות הטבלה יהיו מאוחדים לקו אחד ולא קו כפול.

10. השתמשו בהגדרות id ו-class במידת הצורך.

תרגיל 53

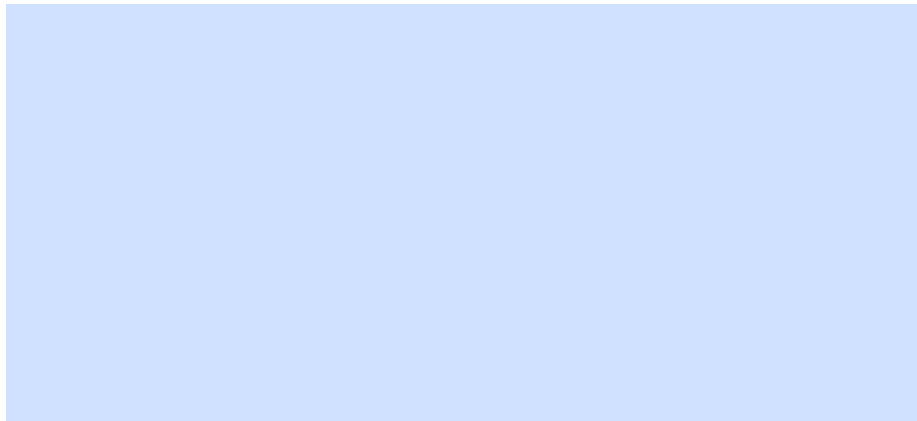
1. העתיקו את הטבלה מהתרגיל הקודם.

2. הוסיפו עמודת "עיר".

3. גון גר בתל אביב.

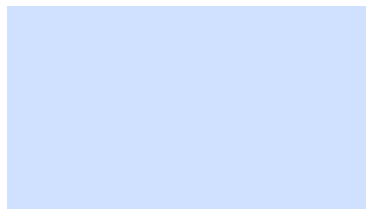
4. גל ואווה גרים בירושלים, הגדירו מיזוג עמודות לתא עיר שלהם.

תצוגת השינוי בשינויים בדפדפן תראה כך:



תרגיל 54

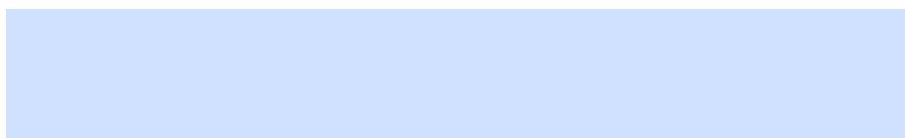
1. צור מסמך html חדש.
2. צור במסמך את מראה הטבלה הבא:



3. הגדר לטבלה רוחב של 50%.

תרגיל 55

1. צור מסמך html חדש.
2. צור במסמך את מראה הטבלה הבא:



3. הגדר לטבלה רוחב של 100%.

טפסים

טפסים

ברב האתרים נרצה לקבל אינפורמציה מהגולש, את האינפורמציה מהגולש נקבל באמצעות טופס, הטופס יכול להכיל אלמנטים שונים: שדות קלט, תיבות בחירה, תיבות בחירה מרובות, אזורי טקסט, תיבות ססמא ועוד...

התגית <form>

את פקדי הטופס עוטפים עם התג <form>, תגית זו פותחת וסוגרת את הטופס, לתגית זו מאפיין חשוב, המאפיין action, מאפיין זה מציין לאן ישלח מידע הטופס, הוא מאפיין חובה בכל טופס פעיל.

דוגמא לטופס בסיסי:

<h2>טופס מיוחד</h2>

<form

name="example" method="get" action="/action_page.php">

שם פרטי:

<input type="text" name="firstname">

שם משפחה:


```
<input type="text" name="lastname">
```

```
<br><br>
```

```
<input type="submit" value="שלח">
```

```
</form>
```

ניתוח הנתונים המתקבלים מהטופס אינו שייך לשפת ה-html, אלא לשפות שרת, ה-html מאפשר לדפדפן לשלוח את הנתונים בשרת, השרת מקבל את הנתונים באמצעות תכנית ייעודית שנכתבה לשם קבלת הנתונים וטיפול בהם, שפות המאפשרות לנתח את הנתונים המתקבלים לשרת הן שפות שרת, כמו לדוגמה שפת php ודומיה.

המאפיין action

נוכל לראות בדוגמה התגית form קיבלה את המאפיין action מאפיין זה הוא חשוב מאד ומגדיר לטופס להיכן הדפדפן ישלח את הנתונים לאחר שילחצו על הלחצן submit, לחצן השליחה.

המאפיין method

מאפיין נוסף חשוב שהוגדר בתגית form, המאפיין method, מאפיין זה מורה לטופס באיזו שיטה נתוני הטופס יעברו, קיימות שתי דרכים להעברת המידע מהטופס:

דרך אחת get, דרך זו היא ברירת המחדל, היא מאפשרת לשלוח כמות מוגבלת של נתונים לשרת. שיטת ה-get מגדירה שהפרמטרים שנקלטים בטופס ומועברים לעמוד הנדרש דרך שורת הכתובת, אם הגדירו את שיטת השליחה בצורה הזו, ברגע שילחצו על הלחצן שלח, נראה את הטקסט של הנתונים שעוברים כתוב בשורת הכתובת בדפדפן.

לדוגמה: נראה לאחר שליחת הטופס נוכל לראות את הנתונים בכתובת כך:

<https://exampositesite.com/example.php?user=something&lastname=something>

את הנתונים המועברים נראה לאחר השליחה בשורת הכתובת כשלאחר שורת הכתובת יבוא התו סימן שאלה (?), לאחר הסימן שאלה נראה את הפרמטרים שהועברו: שם הפרמטר

הסימן שווה (=) ולאחריו ערך הפרמטר, אם הועברו מספר פרמטרים נראה את הסימן & בין פרמטר אחד לשני. בדוגמה הנ"ל אנחנו רואים שהשדה user העביר את הערך something, השדה lastname העביר את הערך something, כלומר גם בשדה firstname וגם בשדה lastname הגולש הזין, הקליד את הערך something.

דרך נוספת להעברת נתונים מהטופס: **post** דרך זו מגדירה שהפרמטרים יועברו ללא שהמשתמש יראה אותם, היא משמשת להעברת רגיש כמו פרטי גולש סיסמאות, נתוני אשראי ושאר פרטים שמעוניינים שלא יחשפו בעת העברתם.

פקדים שניתן להוסיף בטופס

מספר פקדים שניתן להוסיף לטופס על מנת להוסיף בו שדות שונים:

תגיות קלט

בטופס ישנן מספר תגיות שמאפשרות לקבל קלט מהמשתמש, לשדות הקלט מאפיין חשוב בשם name, חשוב מאד להגדיר מאפיין זה לכל פקד ופקד בטופס בכדי שהעמוד שמקבל את המידע יוכל לגשת אליו.

מאפיין בשם **readonly** - מציין על הפקד שמוגדר בו שהוא לקריאה בלבד.

המאפיין **disabled** - מציין על הפקד שמוגדר בו כאינו פעיל.

ההבדל בין פקד שאינו פעיל לפקד המוגדר לקריאה בלבד, הוא בכך שפקד המוגדר כלא פעיל, לא ישלחו נתוניו עם שליחת הטופס, אך פקד שהוא לקריאה בלבד, כן נשלחים נתוניו בשליחת הטופס.

התג <input>

התגית input, יוצרת תיבת קלט המשולבת בטופס על מנת להזין מידע לטופס. תגית זו מאפשרת ליצור מגוון רחב של תיבות קלט שונות, היא אינה צריכה תגית סגירה ופתיחה, אלא בנויה מתגית בודדת, ניתן להגדיר באמצעות המאפיין **type**, את סוג תיבת הקלט, המאפיין **type**, מקבל את אחד מהערכים הבאים:

1. **text** - להגדרת תיבת קלט המקבלת טקסט.

maxlength - לקביעת מספר התווים המקסימאלי שניתן להזין לתיבת הטקסט.

size - לקביעת גודל התיבה.

value - לקביעת ערך ראשוני הנתון בתיבת הטקסט.

דוגמא לתיבת טקסט:

```
<input type="text" name="firstName">
```

בדפדפן היא תראה כך:



2. password - להגדרת תיבת סיסמא, תיבת סיסמא היא תיבת טקסט לכל דבר

וענין, מקבלת את אותם מאפיינים כתיבת טקסט רגילה, אך ההבדל היחיד הוא שהיא המסווה את הקלדת הנתונים במסווה של נקודות או כוכביות.

דוגמא להגדרת תיבת סיסמא:

```
<input type="password" name="password">
```

בדפדפן היא תראה כך:



3. checkbox - תיבת סימון המאפשרת לגולש לסמן גם יותר מערך אחד.

דוגמא להגדרת תיבת סימון:

```
<input type="checkbox" name="check1" checked>
```

```
<input type="checkbox" name="check1">
```

בדפדפן תיבות הסימון תיראנה כך:



4. radio - להגדרת לחצני אפשרויות המאפשרים למשתמש לבחור באפשרות אחת מבין מספר מוגבל של אפשרויות, כאשר מסמנים אפשרות אחת, אפשרות אחרת שהייתה מסומנת תתבטל.

דוגמא להגדרת תיבת סימון המאפשרת בחירת אפשרות בודדת:

```
<input type="radio" name="radio1" checked>
```

```
<input type="radio" name="radio1">
```

בדפדפן התיבות תיראנה כך:



שימו לב: לתיבת checkbox ולתיבת radio ניתן להגדיר את הערך **checked** ערך זה מגדיר את הנתון שמצוין עליו כמסומן.

5. file - תיבת טעינת קובץ, תיבה מיועדת להעלאת קבצים.

דוגמא לתיבה לטעינת קובץ:

```
<input type="file" name="file1">
```

בדפדפן התיבה תראה כך:



ניתן להגדיר לתיבת טעינת קובץ מאפיין size המציין את משקל הקובץ המותר להעלאה, כמו"כ המאפיין accept, מציין איזה סוגי קבצים התיבה תוכל לקבל.

6. button - פקד כפתור לשילוב לחצנים בטופס, פקד זה מקבל את המאפיין value, ערכו של ה-value קובע לכפתור את הטקסט שיוצג עליו.

דוגמא לפקד כפתור:

```
<input type="button" name="button1" value="לחצן">
```

בדפדפן הכפתור יראה כך:



7. reset - במידה והקלדנו, מלאנו נתונים בטופס, ונרצה למחק את כולם בפקודה אחת, נשתמש בכפתור `reset`, כפתור זה מנקה את הטופס מנתוניו.

לדוגמא:

```
<input type="reset" name="reset1" value="ניקוי">
```

בדפדפן הוא יראה ככפתור רגיל:



8. submit - כפתור שליחה, ממוקם בסוף כל טופס, תפקידו לשלוח את הטופס לקובץ המוגדר ב-`action` שבתגית ה-`form`, הקובץ שאוסף ממנו את הנתונים ומטפל בהם.

דוגמא לכפתור שליחה:

```
<input type="submit" name="submit1" value="שלח">
```

בדפדפן כפתור שליחה יראה ככפתור לכל דבר ללא הבדל בנראות.



התג `<textarea>` - אזור טקסט

תג זה מוסיף בטופס אזור להזנת טקסט, מעט ארוך מטקסט של תיבת טקסט, פקד זה נשלב במידה ונרצה לקבל מהגולש מידע מעט יותר מפורט, כמו תוכן פנייה וכדו' בתגית זו המאפיין **rows** יציין את מספר השורות שיוצגו באזור תיבת הטקסט בהקלדת טקסט לתיבה, להצגת שורות נוספות יש צורך בשימוש בפס הגלילה האנכי. המאפיין **cols** יציין את הרוחב של אזור הטקסט, רוחב תיבת הטקסט יצוין במספר תווים, כלומר כמה תווים יכנסו בכל שורה.

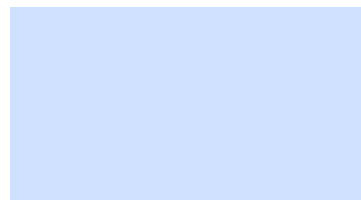
לתגית זו יש תגית פותחת ותגית סוגרת בין התגית הפותחת לסוגרת נמקם את הטקסט הראשוני שנרצה לשים באזור הטקסט.

המאפיין **maxlength** מציין את מספר התווים המקסימאלי שניתן להזין בתיבה זו.

דוגמא לאזור טקסט:

```
<textarea rows="5" cols="20" name="textarea1"
maxlength='150'>הטקסט הראשוני כאן</textarea>
```

בדפדפן אזור תיבת הטקסט יראה כך:



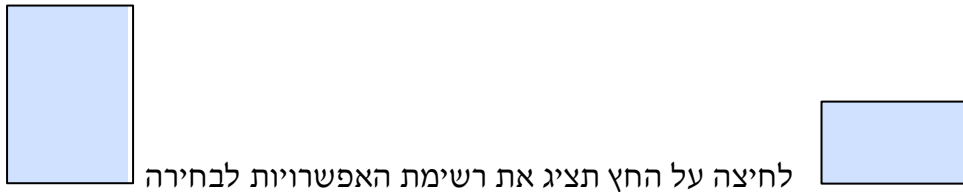
התג <select> - תיבת בחירה

התג `select`, מציין תיבת בחירה בטופס, תג זה בנוי מזוג תגיות: פותחת וסוגרת, בין תגית הפתיחה לתגית הסגירה נמקם את התגיות `<option>`, התג `option` גם כן בנוי מזוג תגיות, כשבין התגית הפותח לסוגרת ימוקם הטקסט שנרצה להציג באפשרות הבחירה, ניתן למקם בתיבה `select` מספר בלתי מוגבל של תגיות `option`, לציון מספר רב של אפשרויות בחירה.

דוגמא לתיבת בחירה:

```
<select name=select1>
  <option value="1">אחד</option>
  <option value="2">שנים</option>
  <option value="3">שלוש</option>
  <option value="4">ארבע</option>
  <option value="5">חמש</option>
</select >
```

בדפדפן תיבת הבחירה תראה כך:



ניתן להגדיר את מספר הערכים הגלויים באמצעות המאפיין size לדוגמא:

```
<select name=select1 size="3" multiple>
```

```
<option value="1">אחד</option>
```

```
<option value="2">שנים</option>
```

```
<option value="3">שלוש</option>
```

```
<option value="4">ארבע</option>
```

```
<option value="5">חמש</option>
```

```
</select >
```

בדוגמא זו שלושה ערכים יהיו גלויים לבחירה כך:



באמצעות המאפיין multiple ניתן לאפשר למשתמש לבחור יותר מערך אחד, על מנת לבחור מספר ערכים, על המשתמש לבחור את הערכים הנוספים בצירוף המקש ctrl. המאפיין multiple נכתב כמו שהוא, הוא מגדיר את האפשרות ואינו מקבל ערך.

התג <datalist> - רשימת נתונים

התג datalist יוצמד לתיבת קלט, אלמנט זה מציין רשימת אפשרויות מוגדרות מראש, ומאפשר למשתמש להשלים את הקלדתו, על ידי סינון האופציות מתוך הרשימה לפי התווים שהמשתמש מתחיל להקליד, ניתן גם לבחור מתוך רשימת האפשרויות ללא הקלדה כלל

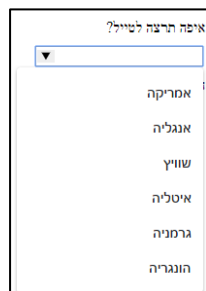
בלחיצה על החץ הנוצר לצד תיבת הקלט, על מנת לקשר בין תיבת הקלט לרשימת הנתונים, יש לתת בתיבת הקלט מאפיין בשם list, המאפיין list יקבל את ה-id של ה-datalist, כמו כן חובה להגדיר id לתיבת ה-datalist.

דוגמא לרשימת נתונים:

```
<input list="States" name="States">  
  
<datalist id="States">  
  
  <option value="אמריקה">  
  
  <option value="אנגליה">  
  
  <option value="שוויץ">  
  
  <option value="איטליה">  
  
  <option value="גרמניה">  
  
  <option value="הונגריה">  
  
</datalist>
```

בדוגמא זו אם המשתמש התחיל להקליד את האות א הוא יקבל את רשימת האפשרויות המתחילות באות א' לבחירה: אמריקה, אנגליה ואיטליה.

בתצוגת הדפדפן תיבת הנתונים תראה כך:



אם נתחיל לכתוב את האות א', נראה את התצוגה כך



ונוכל ללחוץ לבחירת כל אחת מהאופציות המופיעות ברשימה.

התגית <label> לפקדים בטופס

התגית <label> משמשת ליצירת תווית טקסט לצד הפקדים בטופס (input, radio) במקום להשתמש בתגיות אחרת להכללת התוכן לתיאור הפקד.

לדוגמא:

```
<label>שם:</label>
```

```
<input id="firstname" type="text" />
```

וכך זה יראה בדפדפן:



ה-label עוזר להוסיף שם או ערך בסמוך לפקד. בפקדים בהם הערך נקבע באמצעות המאפיין value, אין להוסיף label, כמו לדוגמא בפקד button.

תגית label עומדת על יד כל שדה על מנת לבאר את פשרו ותפקידו, על מנת לקשר בין תגית לשדה אליהם היא קשורה נוסף בתגית label את המאפיין **for**, המאפיין for יקבל כערך את זהות הפקד (ה-id שלו), כתוצאה מהגדרת מאפיין זה ברגע שנלחץ על ה-label, ה-label יקפיץ אותנו לתוך השדה אליו הוא מקושר.

דוגמא נוספת לשימוש ב-label:

```
<form action="/action_page.php" dir="rtl">
```

```
<label for="male">זכר</label>
```

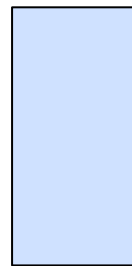
```
<input type="radio" name="gender" id="male" value="male"><br>
```

```

<label for="female">נקבה</label>
<input type="radio" name="gender" id="female" value="female"><br>
<label for="other">אחר</label>
<input type="radio" name="gender" id="other" value="other"><br><br>
<input type="submit" value="שלח">
</form>

```

כך זה יראה בתצוגה בדפדפן:



תגית ה-lable מומלצת מאד להגדרה בטפסים, היא עוזרת להנגשת האתר לבעלי מוגבלויות, קוראי המסך נעזרים בתגית lable, המתארת ומסבירה את מהות הפקד.

תרגול 56

1. צרו מסמך html חדש.
2. צרו במסמך טופס בסיסי.
3. הטופס יכיל את השדות: שם פרטי, שם משפחה, כתובת אימייל והלחצן שלח.
4. שמרו את הטופס, הפעילו אותו בדפדפן לבדיקה שהכל נראה כראוי.

התג <fieldset>

התגית fieldset משמשת אותנו להגדרת אזור למספר אלמנטים בעלי מכנה משותף בתוך טופס, לדוגמא כל הפרטי משתמש, או כל פרטי ההתקשרות, כל פרטי ההשכלה וכדומה, נוכל לאגד בתוך התג fieldset, תג זה מוסיף מסגור לפקדים.

ניתן להגדיר כותרת שתמוקם בתוך המסגור באמצעות התגית legend.

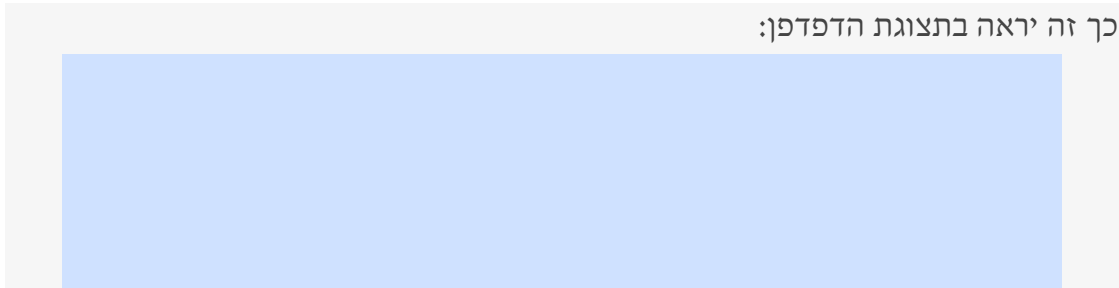
דוגמא לשימוש בתגיות fieldset ו-legend:

```

form dir="rtl">
<fieldset>
<legend>פרטים אישיים:</legend>
שם:<input type="text"><br>
אימייל: <input type="text"><br>
תאריך לידה:<input type="text">
</fieldset>
</form>

```

כך זה יראה בתצוגת הדפדפן:



תרגיל 57

1. פתח מסמך html חדש
2. צור טופס עם 3 אזורים: פרטים אישיים, פרטים על בן הזוג, פרטים על מקום העבודה.
3. כל אזור בטופס יאוגד בתוך התגית fieldset.
4. לכל אזור בטופס הגדירו כותרת בתגית legend.
5. באזור פרטים אישיים הוסף את השדות הבאים: שם פרטי, שם משפחה, תאריך לידה, כתובת, טלפון.
6. באזור הפרטים על בן הזוג הוסף את השדות הבאים: שם פרטי, שם משפחה, תאריך לידה.
7. באזור הפרטים על מקום העבודה הוסיפו את השדות הטקסט הבאים, שם חברה, כתובת, שם מנכ"ל החברה, תפקיד בעבודה.
8. שמרו את המסמך, הציגו בדפדפן ובדקו שהכל עובד כראוי.

המאפיין Place holder - שומר מקום

מאפיין זה הוא כעין שומר מקום לטקסט, הערך שניתן למאפיין זה יוצג בתיבות הטקסט כערך אפרפר, ברגע שהגולש ילחץ על תיבה שהוגדר בה place holder, ה--place

holder יעלם, אם הגולש יזין תוכן, הערך האפרפר לא יופיע שוב, אם הגולש יעבור לתיבה אחרת ללא הזנת תוכן, ה-place holder יופיע שוב עזיבת הפקד.

לדוגמא:

```
<form action="/action_page.php">  
  
<input type="text" name="fname" placeholder="שם פרטי"><br>  
  
<input type="text" name="lname" placeholder="שם משפחה"><br>  
  
<input type="submit" value="שלח">  
  
</form>
```

כך זה יוצג בדפדפן:

תרגיל 58

9. צרו מסמך html חדש.
10. צרו טופס המכיל את השדות הבאים:
11. שדה שם פרטי, בשדה יופיע כ-placeholder, הטקסט "מה השם שלך"
12. שדה שם משפחה, בשדה יופיע כ-placeholder, הטקסט "מה המשפחה שלך"
13. שדה כתובת אימייל, בשדה יופיע כ-placeholder, הטקסט "נשלח רק דברים שווים"
14. שדה טלפון, בשדה יופיע כ-placeholder, הטקסט "בשביל שנוכל לעזור לך"
15. לכל שדה המתאר יופיע בתוך תגית label.
16. שמרו את המסמך, צפו בדפדפן שהכל נראה כראוי.

תיבה מוסתרת

תיבה מוסתרת היא תיבה המשרתת אך ורק את הטיפול מאחורי הקלעים בטופס ולא לתצוגה למשתמש כלל.

לעיתים נרצה להגדיר שערך מסוים ישלח עם נתוני הטופס, מבלי שהמשתמש יראה אותו, למטרה זו ניתן להגדיר תיבת טקסט מסוג תיבה מוסתרת, התיבה תהיה מאותחלת עם ערך שהמשתמש לא יראה אותו ובכך לא יוכל לגעת בו.

תיבה מוסתרת לא תוצג לגולש. נגדיר תיבה לתיבה מוסתרת על ידי מתן הערך `hidden` למאפיין `.type`.

לדוגמא:

```
<input type="hidden" id="custId" name="custId" value="1586">
```

הגדרנו בדוגמא זו שהתיבה מוסתרת, ויש בה את הערך 1586, כשהטופס ישלח את הנתונים המלאים לאחר מילוי, ישלח גם ערך זה.

תרגיל 59

1. צרו מסמך html חדש
1. המסמך יכיל טופס רישום לגן ילדים
2. הוסיפו כותרת "רישום לגן אילנה"
3. הוסיפו טופס עם האזורים הבאים:
4. פרטי הילד השדות באזור:
 - a. שדה שם הילד
 - b. תאריך לידה
 - c. מגדר: תיבת סימון בין האופציות: זכר או נקבה.
 - d. מספר ילדים במשפחה, תיבת בחירה, הנתונים בתיבת הבחירה הם: 1,2,3,4,5,6,7,8,9,10
5. אזור 2: פרטי ההורים השדות באזור:
 - a. שם האב

- b. שם האם
- c. כתובת
- d. טלפון
- 6. בתחתית הטופס מקמו לחצן לניקוי הטופס, ולחצן שלח.
- 7. שמרו את הקובץ.
- 8. פתחו את הקובץ בדפדפן ובדקו שהכל עובד כראוי.

תרגיל 60

- 1. צרו מסמך html חדש
- 2. הוסיפו כותרת טופס רישום למכללה
- 3. צרו במסמך טופס רישום למכללה
- 4. הטופס יכלול את השדות הבאים:
 - a. שם מלא
 - b. מגדר: תיבת סימון בין האופציות זכר או נקבה.
 - c. כתובת
 - d. טלפון
 - e. שנת לידה
 - f. שדה השכלה: תיבה נפתחת עם הערכים: תיכונית, על תיכונית, תואר ראשון, תואר שני, דוקטורט.
 - g. טופס רישום למגמה: תיבה נפתחת עם הערכים: גרפיקה, מולטימדיה, עיצוב פנים, צילום, הנדסאי חשמל, הנדסאי תוכנה.
- 5. הוסיפו לחצן שלח, ולחצן ניקוי טופס.
- 6. שמרו את הטופס, צפו בדף ובדקו שהכל עובד כראוי.

תרגיל 61

- 1. פתחו מסמך html
- 2. הוסיפו את הכותרת: "המטיילים"
- 3. בנו טופס יצירת קשר, הגדירו בטופס את הפקדים הבאים:
 - a. שם פרטי, תיבת קלט

- b.** שם משפחה, תיבת קלט
- c.** תיבת סיסמא
- d.** הוסיפו את הכיתוב "מה תרצה לשתות"
- e.** הוסיפו 3 תיבות בחירה לבחירת משקאות (קפה, תה, קפוצ'ינו), אפשרו לסמן מספר אפשרויות.
- f.** הוסיפו תיבת רשימה לבחירת אזור מגורים, תיבת הרשימה תאפשר לבחור באחת מבין האופציות הבאות: צפון, דרום, מרכז, ירושלים, אילת.
- g.** הוסיפו את הטקסט "היכן תרצה לטייל".
- h.** הוסיפו תיבת קלט, עם רשימת נתונים לבחירה, הוסיפו בתיבה מספר מקומות לטיול כרצונכם.
- i.** הוסיפו את הכיתוב "מגדר", הוסיפו בחירה המאפשרת סימון אחד, זכר או נקבה.

המטיילים

שם פרטי:

שם משפחה:

סיסמא:

מה תרצה לשתות?

קפה
 תה
 קפוצ'ינו

זכר
 נקבה

איפה תרצה לטייל?

העלה צילום תעודת זהות:

הערות:

הטקסט הראשוני כאן

אזור מגורים ▼

- j.** הוסיפו את הטקסט: "הערות"
- k.** הוסיפו אזור טקסט לכתובת טקסט חופשי.
- l.** הוסיפו את הטקסט "העלה צילום תעודת זהות"
- m.** הוסיפו תיבה לטעינת קובץ.
4. כל פקד בטופס יעמוד בשורה נפרדת
5. קבעו מרווחים של 10 פיקסלים בין שורה לשורה.
6. קבעו לטופס צבע רקע כרצונכם.
7. קבעו מסגרת מסגנון וצבע רק כרצונכם.
8. שמרו את הקובץ ובדקו בדפדפן שהדף עובד כראוי.

אלמנטים לקיבוץ תוכן div ו-span

אלמנטים לקיבוץ תוכן div ו-span

שני אלמנטים שאינם מיועדים לתוכן מסוים, אך משתמשים בהם שימוש נרחב, הם משמשים לקיבוץ מספר אלמנטים יחדיו.

התגית <div> והתגית וההבדל ביניהן

התגיות div ו-span מאפשרות לנו לתת לאזורים שונים בדף עיצוב שונה.

התגית <div> תגדיר אזורים שמקבצים תוכן, כמו מספר כותרות, מספר פסקאות, מספר קישורים מספר תמונות, או מספר אלמנטים מסוגים שונים יחדיו, תגית זו היא תגית הנכנסת להגדרת תגית **block** "בלוק", תגית התופסת את מלוא רוחב הדף, 100% רוחב, ועם הגדרתה, נוצר מעבר שורה לשורה חדשה.

דוגמאות לשימוש בתגית div:

```
<div>
```

```
<a>קישור לדף באתר</a>
```

```
<a>קישור לדף נוסף באתר</a>
```

```
<a>קישור לדף נוסף באתר</a>
```

```
<a>קישור לדף נוסף באתר</a>
```

```
</div>
```

```
<div>
```

```
<h1>זו כותרת למאמר</h1>
```

```
<h2>זו כותרת המשנה למאמר</h2>
```

</p> זו פסקה במאמר שמכילה תוכן רב<p>

</p> זו פסקה נוספת במאמר שמכילה תוכן רב<p>

</p> זו פסקה נוספת במאמר שמכילה תוכן רב<p>

</div>

**התגית ** מגדירה אזורים בתוך הפסקה, נשתמש בה כאשר יש מספר תווים, מילה בודדת או יותר, שאנו מעוניינים להעניק להם עיצוב שונה משאר החלקים בפסקה, תגית זו נכנסת להגדרת תגית inline (בתוך השורה), תגיות inline, הן תגיות שלא יוצרות מעבר שורה וניתן למקם אותן זו לצד זו, באותה השורה.

דוגמא לשימוש בתגית span:

כל המספרים בטקסט שמונה זו פסקה מספר <p>
</p> נרצה שיהיו בעיצוב מיוחד

הגדרנו את התגית span ובתוכה המילה "שמונה", ברצוננו לתת עיצוב מיוחד למילות המגדירות מספר, כעת נוכל להתייחס ספציפית למילה שמונה, ושאר המספרים שנשלב בטקסט בצורה כזו שנתייחס לקלאס number ששילבנו בתגית span, אם לא ניתן הגדרת עיצוב מיוחדת לspan שהוגדר, לא יראה כל הבדל בין המילה "שמונה" לשאר חלקי הפסקה, כל המטרה בשימוש בתגית זו היא לבדל חלק מסוים מהפיסקה על מנת שתהיה האפשרות לתת עיצוב ספציפי לחלקים המבודדים.

לסיכום ההבדל בין div ל-span

1. התגית span היא תגית Inline - בתוך השורה נועדה לעיצוב בעיקר של חלק מתוך פסקה.
2. התגית div נועדה לקבץ יותר אלמנטים, ומשמשת כבלוק - תופסת את כל רוחב העמוד.

החל מ-`html` גרסא 5 נוספו עוד מספר אלמנטים נוספים המסייעים לארגון וקיבוץ תוכן ומקבצים חלקים שונים במסמך, ובנוסף שמם מעיד על תפקיד האלמנטים המקובצים בהם ועל מיקומם במסמך, כמו לדוגמא התגית `<nav>` (קיצור של המילה `navigation` - ניווט) שתפקידה לאגד קישורי ניווט באתר, התגיות הללו הן תגיות בעלות משמעות סמנטית, בעוד שהן מארגנות את המסמך שיראה טוב לעיני הגולש, הן מארגנות גם את המסמך לדפדפן ומנועי החיפוש ומעידות על תוכנו, התנהגותם של התגיות הסמנטיות החדשות זהה להתנהגות של התגית `div`, הם בעצם כעין `div` בעלי שם שונה.

חלק מהתגיות הן:

header

תגית זו מגדירה כותרת של דף או של חלק מדף. בדרך כלל היא מופיעה בראש הדף, אך ניתן למקמה כמעט בכל מקום בדף, במקרים רבים היא כוללת בתוכה תפריט עליון, תמונת לוגו האתר וכדו'. תגית זו יכולה לכלול כמעט כל תגית פרט לתגיות `<header>` ו-`<footer>`, ניתן גם להגדיר מספר תגיות `header` באותו מסמך.

דוגמא:

```
<header>
  <h1>?מתי עלתה החללית הראשונה לחלל</h1>
  <p>המרוץ לחלל</p>
</header>
```

footer

תגית זו משמשת להגדרת כותרת תחתונה למסמך או לקטע במסמך, בתגית זו יופיעו בד"כ דברים פחות חשובים, ציון אזהרת זכויות יוצרים, ציון שם מחבר או בונה האתר, קישורים למפת האתר ולחלקיו וכדומה.

דוגמא:

```
<footer>
  <p>פורסם על ידי: אנה דויד</p>
```

```
<p><a href="mailto:someone@example.com">someone@example.com</a></p>
</footer>
```

<article>

תגית זו מגדירה מאמר שיכול להיות גם עצמאי בפני עצמו ככתבה שיכולה להופיע ככתבה בפני עצמה, תוכן הממוקם בתגית זו, הוא תוכן שנוכל לקחת אותו בשלמותו ולהניח אותו במקום אחר, ועדיין לתוכן תהיה משמעות משלו.

לדוגמא:

```
<article>
```

```
<h1>גוגל כרום</h1>
```

```
<p>גוגל כרום שפורסם בשנת 2008, הוא דפדפן אינטרנט חנימני וקוד פתוח שפותח על ידי גוגל כרום</p>
```

```
</article>
```

nav

תגית זו משמשת אותנו להגדרת אזור המכיל קישורים שונים לניווט כמו תפריט, או קישורים לרשתות חברתיות, או כל קבוצת קישורים חשובים, לא נשתמש בתגית זו עבור קישור בודד.

דוגמא:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<nav>
```

```
<a href="#">HTML</a> |
```

```
<a href="#">CSS</a> |  
<a href="#">JavaScript</a> |  
<a href="#">jQuery</a>  
</nav>  
</body>  
</html>
```

<main>

תגית זו מיועדת לאזור התוכן המרכזי בדף. תגית זו מופיע פעם בודדת בדף ולא יכולה להופיע יותר מכך, תגית זו לא יכולה להיות בתוך התגיות article, aside, header, footer, nav כמו כן התגית תכיל את התוכן המרכזי ולא מומלץ להכניס בתוכה אלמנטים שחוזרים על עצמם בכל עמוד באתר כמו תפריט ניווט ראשי וכדומה.

דוגמא:

```
<main>  
<h1>דפדפני אינטרנט</h1>  
<p>הם דפדפנים הנפוצים ביותר Google Chrome, Firefox,Internet Explorer כיום</p>  
  
<article>  
<h1>גוגל כרום</h1>  
<p>הוא דפדפן אינטרנט חנימי וקוד פתוח שפותח על ידי Google Chrome גוגל כרום</p>  
</article>  
  
<article>  
<h1>אינטרנט אקספלורר</h1>  
<p>הוא דפדפן אינטרנט חנים ממיקרוסופט, Internet Explorer אינטרנט אקספלורר</p>  
</article>
```


</article>

<article>

<h1>מזילה פיירפוקס</h1>

<p>פיירפוקס הוא דפדפן אינטרנט ללא קוד פתוח ממוזילה, שיצא בשנת 2004.</p>

</article>

</main>

<aside>

תגית זו מגדירה תוכן צדדי יחסית, מתאים להערת שוליים, תפריט צדדי או כותרות צדדיות במאמר.

לדוגמא:

<aside>

<h4>פארק המרמולדה</h4>

<p>הוא פארק מלא שעשועים לילדים בבאר שבע</p>

</aside>

section

התג משמש להגדרת קטע או חלק מדף כשמדובר בתוכן שלא עומד כשלעצמו, למשל: חלק אחד לחדשות וחלק נוסף לאירועים.

figure

תגית זו משמשת להגדרת תכונות עצמאיות, כגון איורים, דיאגרמות, תמונות, רישומי קוד וכדו'.

<figure>

</figure>

time

להגדרת תאריך/ שעה

לדוגמא:

```
<p>כל יום<time>10:00</time> אנחנו פותחים בשעה </p>
```

תרגיל 62

1. צרו מסמך html חדש.
2. חפש בויקיפדיה את הערך "חלל", או פתח את התיקייה "חלל".
3. העתק את המאמר על המרוץ אל החלל, חלק את המאמר לאזורים בתגיות מתאימות.

תרגיל 63

1. צרו מסמך html חדש.
2. צרו במסמך 4 אזורי div.
3. קבעו לכל אזור צבע רקע שונה

תרגיל 64

1. העתיקו את התוכן הבא:

יום ראשון, הוא היום הראשון בשבוע

יום שני, הוא היום השני בשבוע

יום שלישי, הוא היום השלישי בשבוע

יום רביעי, הוא היום הרביעי בשבוע

יום חמישי, הוא היום החמישי בשבוע

יום שישי, הוא היום השישי בשבוע

שבת הוא היום השביעי בשבוע

ימים שני, שלישי, שישי ושבת מתחילים באות **ש'**

יום חמישי מתחיל באות **ח'**

ימים ראשון ורביעי מתחילים באות ר'

האותיות של כל הימים יחד מרכיבות את המילה שחר

2. הניחו את המילים והאותיות המודגשות בתוך התגית span.
3. הגדירו באמצעות CSS לכל המילים הממוקמות בתגית span, הדגשה וקו תחתי
4. קבעו שכל המילים הממוקמות ב-span תקבלנה עיצוב בבצע תכלת.
5. עטפו את כל הטקסט בתגית div.
6. הגדירו לתגית div מסגרת כרצונכם.
7. הגדירו לתגית div צבע רקע כרצונכם.
8. נסו למרכז את אלמנט ה-div למרכז.
9. קבעו שכל הטקסטים ימורכזו.

תרגיל 65

1. צרו מסמך html חדש.
2. הוסיפו במסמך את הכותרת "גינה יפה ונחמדה"
3. הוסיפו 5 תמונות של צמחים
4. הגדירו לתמונות גבולות
5. אגדו את כל התמונות בתגית div.
6. הוסיפו תחת התמונות קישורים ל3 אתרים שונים.
7. הכניסו את הקישורים בתגית מתאימה.

תרגיל 66

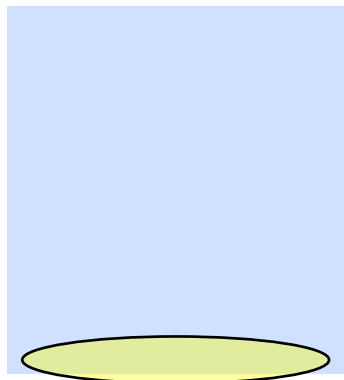
1. פתחו מסמך html חדש
2. כתבו את המשפטים הבאים והשלימו אותם:
 - a. נולדתי בתאריך _____
 - b. עכשיו השעה _____ בערב
 - c. אתמול היה תאריך _____
 - d. בתאריך _____ טסנו ליוון
3. מקמו כל משפט בתגית p. הגדרות תאריך ושעה הכניסו בתגית המתאימה.

בדוק מרכיב, התכונה `display`, התכונה `position`

המשך CSS

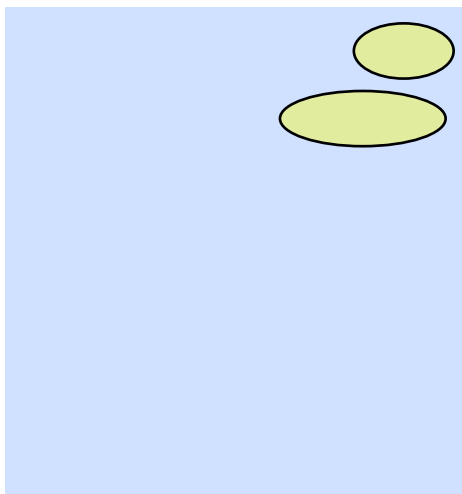
הכרות עם בדוק מרכיב - `f12`

כאשר נגלוש לאתר כלשהו ברשת, נוכל לעמוד על הדף באתר ללחוץ לחיצה ימנית בעכבר ולבחור באפשרות "בדוק" או "בדוק מרכיב",



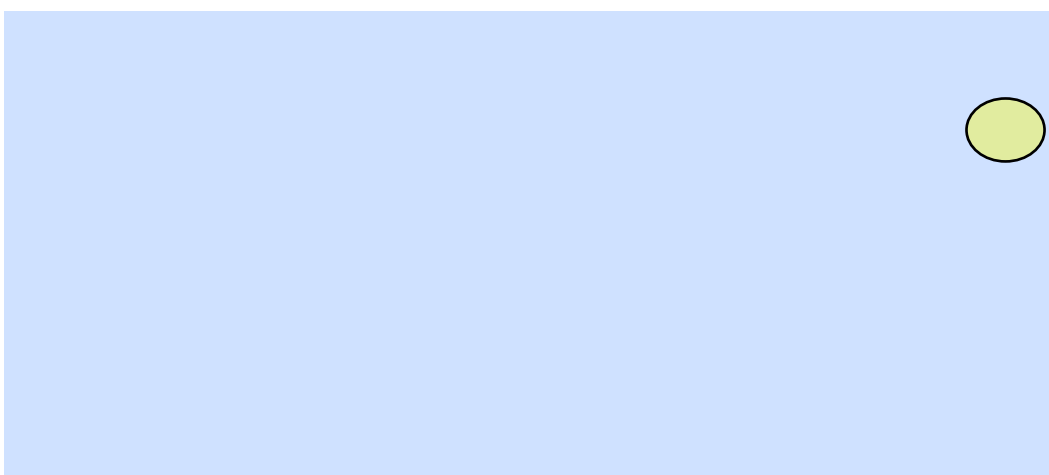
באמצעות ה"בדוק מרכיב" נוכל ללמוד על פקודות ה-CSS המשפיעות על תצוגת האתר, באמצעות כלי זה אנחנו יכולים "לחקור" את העמוד בו אנו צופים, ללמוד על המאפיינים `class` ו-`id` השתולים באלמנטים השונים בעמוד, נוכל ללמוד על כל אלמנט איזו פקודות CSS משפיעות עליו וגורמות לו לעיצוב כזה או אחר.

בלחצן ... (שלוש נקודות)



נוכל לבחור מיקום וצורת פריסה לחלונית ה"בדוק מרכיב".

באמצעות הלחצן + (פלוס) המופיע בצד ימין, נוכל להוסיף סלקטור ל-CSS, ולהוסיף הגדרות CSS שונות.



נוכל כביכול "להוסיף" נתונים בדף, להוסיף סלקטורים ופקודות CSS משלנו לשינוי עיצובו, כמובן כל השינויים שאנחנו מבצעים באתר, אינם ממשיים ורק נראים כל עוד שאנחנו במצב העריכה עם חלונית בדוק מרכיב, ונראים אך ורק בתצוגת הדפדפן במחשב שלנו, אם נרענן את העמוד, השינויים ימחקו והוא יחזור לתצוגתו המקורית כפי שהיא מופיע באתר החי. בפקודות ה-CSS לעיתים רבות נראה פקודות עם סימן קו מחיקה עליהם, פקודות אלו הן פקודות CSS הנדרסות מפקודות CSS חזקות יותר.

יתכן שנרשום פקודת CSS משביעת רצון, וכשניגש לאתר ונטמיע את אותה פקודה, היא לא תשפיע כשם שהיא השפיע בתצוגת ה"בדוק מרכיב", יתכן שדבר זה יקרה מסיבת סדר קדימות CSS ומחוזק של סלקטורים אחרים שגוברים.

בחלונית ה"בדוק מרכיב" נראה גם את תגיות ה-html, נוכל "כביכול" למחוק כך נתונים מהדף, או "להוסיף" נתונים לדף, כמובן כל התוספות למיניהן, הן תוספות שיוצגו רק בדפדפן במחשבנו.

תרגיל 67

1. גשו לאתר כלשהו.
2. צרו צילום מסך (המקשר prtscn במקלדת) של התצוגה הנראית לעינכם..
3. הדביקו את צילום המסך בקובץ וורד (ctrl+v)..
4. שנו את צבע הטקסט של אחת מהפסקאות בעמוד.
5. מחקו את אחד מהאלמנטים המופיעים בעמוד. (תמונה וכדומה).
6. הוסיפו לאחד מהאלמנטים class בשם my-mew-class, נסו להתייחס לקלאס זה ולהוסיף לו פקודת CSS שתעצב אותו באופן חדש.
7. צרו צילום מסך של הדף בתצוגתו "החדשה".
8. הדביקו את הצילום מסך בקובץ הוורד לצד צילום המסך של האתר המקורי.

הפקודה **!important**

לעיתים ננסה לשנות עיצוב של אלמנט באתר, אך לאחר ניסיונות חוזרים ונשנים נראה שפקודות ה-CSS שאנו מגדירים אינן משפיעות על המראה של האלמנט הרצוי, תופעה זו נובעת מכך שקיימים סלקטורים חזקים יותר המשפיעים על האלמנט אותו אנו מנסים לעצב, או עניין של סדר קדימויות במיקום פקודות ה-CSS, במקרים אלו ניתן להשתמש בפקודה **!important**, פ

פקודה זו היא פקודה שמומלץ מאד להיזהר ממנה, ולהתרחק ממנה במידת האפשר. ברגע שהשתמשנו פעם אחת ב**!important** נאלץ להשתמש בה שוב ושוב ללא מוצא, כדי לחרוג

מהגדרות CSS קודמות וככל שהשימוש בה רחב יותר, יהי קשה יותר לעקוב אחרי הקוד ולהבין את תוצאתו הצפויה.

מומלץ להשתמש בה אך ורק במקרים בהם אנו משלבים בקוד שלנו קוד חיצוני ולא מצאנו דרך אחרת לגרום לCSS שלנו להשפיע.

הפקודה תופיע בצמוד להגדרת ה-CSS אותה אנו מעוניינים לחזק לפני הסימן המפריד פסיק נקודה (;) לדוגמא:

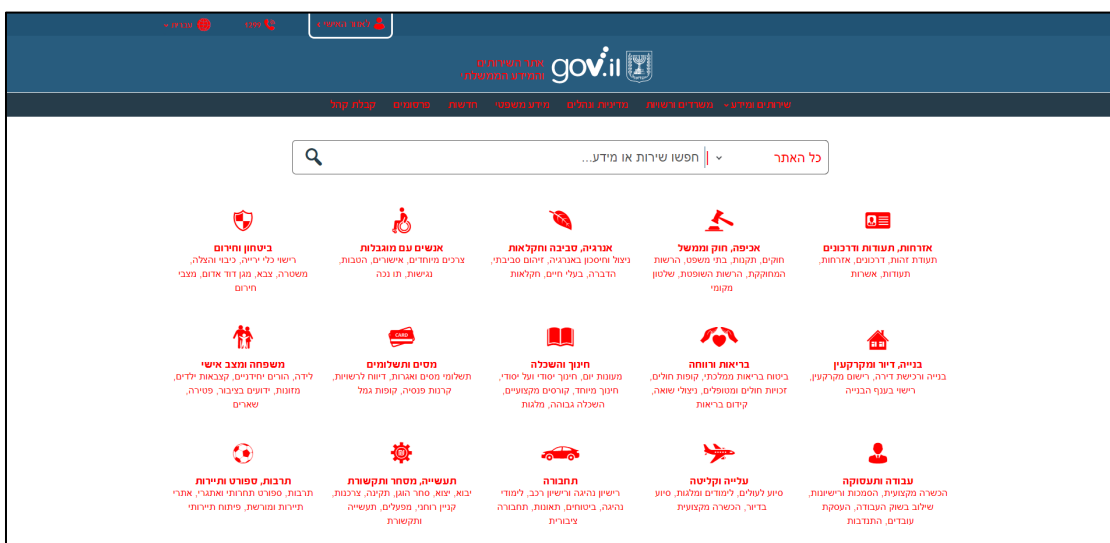
```
Span { color: red !important; }
```

תרגיל 68

1. גלשו לאתר השירותים והמידע הממשלתי: <https://www.gov.il/>
2. נסו לשנות את צבע הטקסט באתר באמצעות הפקודה:
3. `{ color:red;}`
4. אם חלק מהטקסטים לא קיבלו את השינוי הוסיפו את ההוראה `!important`.



התוצאה תראה כך:



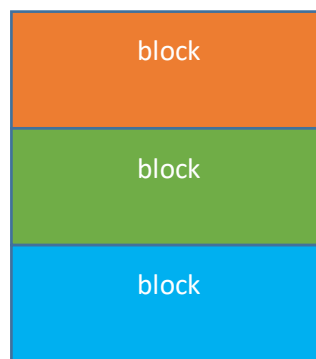
התכונה Display - צורות תצוגה

התכונה display משפיע על אופן הצגתם של אלמנטים, אם האלמנטים יעמדו זה לצד זה או זה תחת זה וכדו'.

הערכים האפשריים שהתכונה display מקבלת הם:

1. **Block** - יגרום לאלמנט להופיע לבדו, לתפוס את כל רוחב השורה של האלמנט.

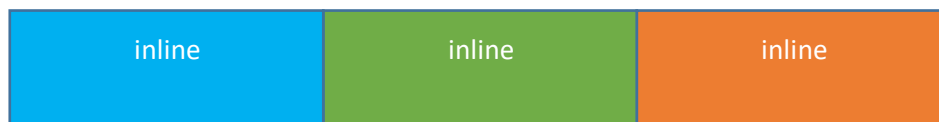
הגדרה זו למספר אלמנטים, תגרום לאלמנטים להופיע זה תחת זה כך:



גם אם מדובר באלמנטים שבבסיסם הם אלמנטים שאינם תופסים את רוחב השורה ויכולים להופיע זה לצד זה במקורם, התכונה block תהפוך אותם לאלמנטי בלוק שיופיעו אחד תחת השני.

במידה ונרצה שהאלמנטים יעמדו זה לצד זה, נשתמש בערך:

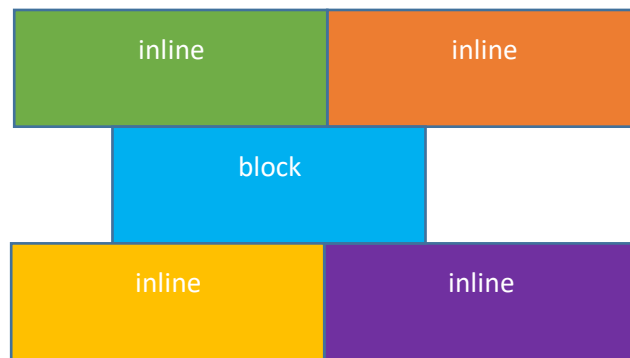
2. **Inline** - ערך זה גורם לאלמנטים להופיע אחד לצד השני באותה שורה כך:



גם אם מדובר באלמנט שבמקורו הוא אלמנט block, הפקודה תבטל לו את תכונת ה-block ותגרום לו להופיע בשורה.

אם נשלב מספר אלמנטים ולחלקם ניתן הגדרת block ולחלקם הגדרת inline, כך מקום שיופיע אלמנט בעל תכונת block האלמנט יתחיל שורה חדשה ויתפוס את כולה.

לדוגמא:



אלמנטים המוגדרים בהגדרה `display:inline`, לא ניתן לקבוע את רוחבם באמצעות התכונה `width`. על מנת שנוכל לקבוע את רוחבם של אלמנטים נשתמש בפקודה `inline-block`.

3. Inline-block - ערך זה גורם לאלמנטים לעמוד בשורה אחת, אך מאפשר לתת לאלמנטים מאפייני רוחב וגובה שיוצגו בפועל, כך:

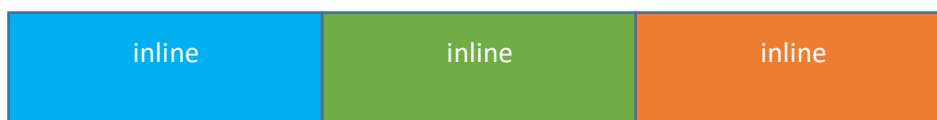
```
.box {  
  
    display: inline-block;  
  
    width: 200px;  
  
    height: 200px;  
  
}
```

4. none - ערך זה יגרום לאלמנט לא להופיע כלל, משתמשים בערך זה לעיתים כשרוצים להעלים, להסתיר אלמנטים, כמו לדוגמא תפריט בנוי בצורה כזו שהאלמנטים שאינם מוצגים כל הזמן, מוגדרים כ-`display:none`, וכשמעוניינים להציג אותם, במעבר עכבר לדוגמא (יפורט בהמשך איך עושים זאת), הם הופכים ל-`display:block`, כך שהם הופכים למוצגים.

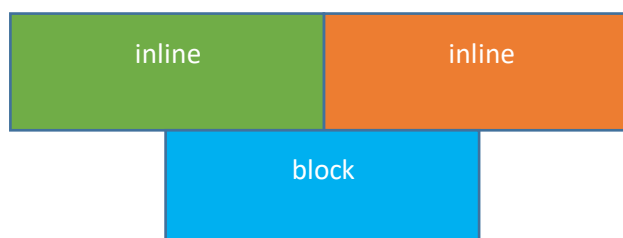
קיימת פקודה נוספת שתפקידה להעלים אלמנטים, הפקודה: `visibility:hidden`. פקודה זו כשהיא מוגדרת היא מעלימה את תצוגת האלמנט, אך מקומו נשמר במסך, בתצוגת ה-`html`, כך שנוצר כעין "חור" במקומו של האלמנט המוסתר, בניגוד לכך הפקודה

`display:none` מעלימה את האלמנט, אך שאר האלמנטים במסמך נדחפים ו"תופסים" את מקומה.

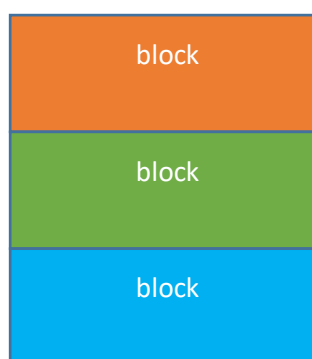
1. צור מסמך html חדש.
2. הכנס בגוף הדף שלוש תגיות div.
3. צור מסמך CSS חדש.
4. בעזרת פקודות CSS, צור את המראה הבא:



5. צור מסמך CSS נוסף, הגדר בו פקודות מתאימות כך שיצרו את המראה הבא.



6. צור מסמך CSS נוסף, הגדר בו פקודות מתאימות כך שיצרו את המראה הבא:



7. הוסף פקודה שתסתיר את ה-div האמצעי (הירוק בתמונה), כך שה-div שאחריו יתפוס את מקומו.
8. אם אנו רוצים ליצור "חור" ולשמור את מקומו של ה-div המוסתר, באיזו פקודה נשתמש? _____

תרגיל 69

1. צרו מסמך html חדש.
2. הוסיפו במסמך 5 פסקאות.
3. הגדירו לכל פסקה צבע רקע שונה.
4. הגדירו שחמשת הפסקאות, תופענה זו לצד זו.

תרגיל 70

1. פתחו קובץ חדש.
2. הוסיפו 3 תמונות נוף.
3. הגדירו ששלושת התמונות, תעמודנה זו תחת זו.

תרגיל 71

1. הוסיפו 5 קישורים.
2. הוסיפו לקישורים צבע רקע.
3. הוסיפו לקישורים מסגרת בעיצוב כרצונכם.
4. הגדירו שהקישורים יעמדו, זה תחת זה.
5. הסתירו את אחד מהקישורים.

התכונה **position** - מיקום

ניתן לקבוע לאלמנטים במסמך מיקום מסוים.

התכונה מקבלת את אחד מהערכים הבאים:

1. **static** - מיקום סטטי, זוהי ברירת המחדל של האלמנט, מיקומו הטבעי לפי סדר

זרימת האלמנטים הטבעית בדף, במידה ולא הגדרנו מיקום, מיקום האלמנט בדף

מוגדר ככזה, במקומו הטבעי.

יתכן והגדרנו מיקום שונה, ומסיבה כלשהי או במקרה מסוים נרצה להחזיר את האלמנט

למקומו הטבעי, במקרה כזה נשתמש בתכונה כך:

```
Img { position: static; }
```

2. **relative** - מיקום יחסי, אם הגדרנו ערך זה נוכל לקבוע לאובייקט מיקום יחסי,

ביחס למיקום בו הוא היה ממוקם בדף באופן טבעי. את מיקומו נקבע באמצעות

התכונות:

top - (חלק עליון) לציון מספר פיקסלים למטה מהמיקום הרגיל ימוקם האלמנט (מרחק

מהחלק העליון).

bottom - (תחתית) לציון מספר פיקסלים למעלה מהמיקום הרגיל בו ימוקם האלמנט

(מרחק מהחלק התחתית).

right - (ימין) לציון מספר פיקסלים שמאלה מהמיקום הרגיל ימוקם האלמנט (הערך

מגדיר את מידת ההתרחקות מצד ימין, כך שנדחף שמאלה).

left - (שמאל) לציון מספר פיקסלים ימינה מהמיקום הרגיל ימוקם האלמנט. (הערך מגדיר

את מידת ההתרחקות מהצד השמאלי, כך שנדחף ימינה)

לדוגמא:

```
Img { position: absolute; left: 10px; }
```

קבענו שהאובייקט יהיה בעל מיקום יחסי, קבענו לו מיקום הסתה שמאלה של 10 פיקסלים מהמיקום בו הוא היה במקור.

3. absolute - מיקום אבסולוטי, אובייקט שהגדרנו אותו כאבסולוטי, אינו נמצא בסדר האלמנטים הרגיל, אלא כביכול חי בעולם משלו. ויכול לקבל מיקום בכל מקום בדף באמצעות התכונות: `top`, `bottom`, `right`, `left`. המיקום נקבע ביחס לחלון כולו (או לאובייקט עליון שעוטר אותו, המוגדר כ-`relatives`).

לדוגמא:

```
Img { position: absolute; right: 0; top: 0; }
```

הגדרה כזו תגדיר לתמונה שתיצמד לפינה הימנית העליונה של הדף, אם היא ממוקמת בתוך אלמנט עוטר שמוגדר כ-`relative`, היא תיצמד לפינתו הימנית העליונה של האלמנט.

אלמנט שקיבל `position: absolute`, יכול לעלות על אלמנטים אחרים בדף, כיוון שהוא חי בעולם משל עצמו, ללא יחס לשאר האלמנטים בדף.

4. fixed - מיקום קבוע, אובייקט שהגדרנו אותו כ-`fixed` מתנהג כאובייקט אבסולוטי, אך שונה בכך שנשאר במקומו קבוע גם בגלילת הדף.

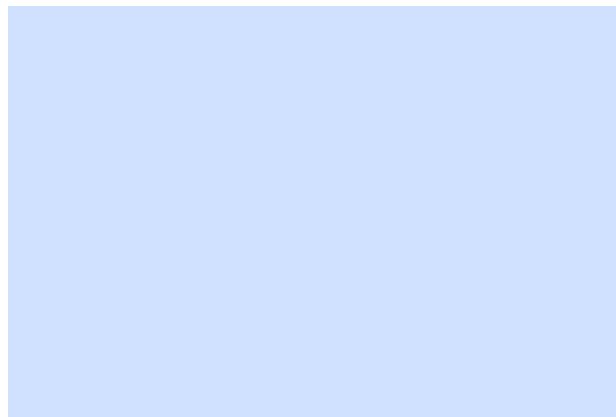
לדוגמא:

```
Img {position: fixed; ; right: 0; top: 0; }
```

הוראה זו קובעת את התמונה ומקבעת אותה לפינה הימנית העליונה של הדף, התמונה תישאר שם גם כשגוללים את המסמך.

1. בחרו ב3 בעלי חיים.
2. חפשו בויקיפדיה, תאור קצר לכל אחד מבעלי החיים שבחרתם או תארו בעצמכם בכמה מילים.
3. הוסיפו 3 אלמנטי div.
4. בכל אלמנט div הוסיפו כותרת עם שם בעל החיים, וטקסט עם תיאור בעל החיים.
5. קבעו מסגרת בעובי של 2 פיקסלים לאלמנטי div, בחרו בצבע כרצונכם.
6. קבעו לאלמנטי ה-div רוחב של 250 פיקסלים וגובה של 500 פיקסלים.
7. הוסיפו אלמנט span, כתבו בתוכו את המילים "קרא עוד".
8. קבעו לתגית ה-span גובה של 50 פיקסלים ורוחב של 100 פיקסלים.
9. הגדירו שהמילים קרא עוד תצמדנה לפינה השמאלית של ה-div.
10. קבעו ששלושת ה-div יופיעו בשורה, זה לצד זה.

דוגמא לתוצאה:



פתרון התרגיל:

```
<!DOCTYPE html>
<html>
<head>
<title>עיצוב דף</title>
```

```

<style>
    p {color: red;}

    div{border: 2px solid red; width:250px; height:500px;
position:relative; display:inline-block; }

    span{position: absolute; bottom:0; left:0; background-
color: yellow; height:50px; width:100px;}

</style>

</head>

<body dir="rtl">

    <div class="two">

        <h2>אריה</h2>

        <p>הוא מין של טורף גדול מהסוג פנתר (Panthera leo: שם מדעי) אריה של
שבמשפחת החתוליים, והוא השני בגודלו במשפחה זו אחרי תת-המין הסיבירי של
האריה נפוץ בעיקר ביבשת אפריקה, אך גם במדינת גוג'אראט שבהודו. הוא ניזון
הטיגריס מאוכלי עשב שונים אותם הוא צד, וכן מפגרים. בתרבות האנושית נחשב
האריה סמל לגבורה וכוח, ולכן הוצמד לו הכינוי "מלך החיות". תוחלת חייו בטבע היא
בין 12 ל-18 שנים, אם כי בשבי הוא יכול לחיות עד 30 שנה. שמות נוספים של
האריה בשפה העברית הם ליש, שחל, שחץ ולביא.[2] שם נקבת האריה הוא לביאה וגור
האריות נקרא כפיר

        </p>

        <span>עוד קרא</span>

    </div>

    <div class="two">

        <h2>אריה</h2>

```


הוא מין של טורף גדול מהסוג פנתר (*Panthera leo*: שם מדעי) אריה<p>
שבמשפחת החתוליים, והוא השני בגודלו במשפחה זו אחרי תת-המין הסיבירי של
האריה נפוץ בעיקר ביבשת אפריקה, אך גם במדינת גוג'אראט שבהודו. הוא ניזון. הטיגריס
מאוכלי עשב שונים אותם הוא צד, וכן מפגרים. בתרבות האנושית נחשב האריה סמל לגבורה
וכוח, ולכן הוצמד לו הכינוי "מלך החיות". תוחלת חייו בטבע היא בין 12 ל-18 שנים, אם כי
בשבי הוא יכול לחיות עד 30 שנה. שמות נוספים של האריה בשפה העברית הם ליש, שחל,
שחץ ולביא.[2] שם נקבת האריה הוא לביאה וגור האריות נקרא כפיר

</p>

קרא עוד

</div>

<div class="two">

<h2>אריה</h2>

הוא מין של טורף גדול מהסוג פנתר (*Panthera leo*: שם מדעי) אריה<p>
שבמשפחת החתוליים, והוא השני בגודלו במשפחה זו אחרי תת-המין הסיבירי של
האריה נפוץ בעיקר ביבשת אפריקה, אך גם במדינת גוג'אראט שבהודו. הוא ניזון. הטיגריס
מאוכלי עשב שונים אותם הוא צד, וכן מפגרים. בתרבות האנושית נחשב האריה סמל לגבורה
וכוח, ולכן הוצמד לו הכינוי "מלך החיות". תוחלת חייו בטבע היא בין 12 ל-18 שנים, אם כי
בשבי הוא יכול לחיות עד 30 שנה. שמות נוספים של האריה בשפה העברית הם ליש, שחל,
שחץ ולביא.[2] שם נקבת האריה הוא לביאה וגור האריות נקרא כפיר

</p>

קרא עוד

</div>

</body>

</html>

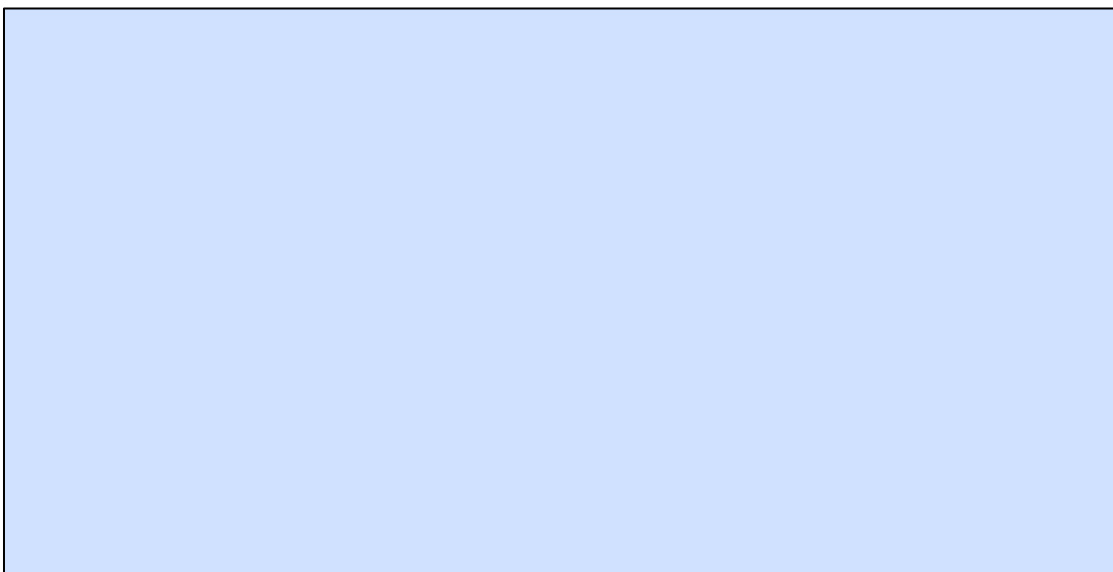
תרגיל 73

1. צרו מסמך html חדש.
2. מלאו את המסמך במלל, ניתן להעתיק או לכתוב מעמכם, נסו למלא כמה שיותר מלל על מנת שיהיה צורך בגלילה בכדי לקרוא את כל החומר.
3. הגדירו קישור למספר טלפון לחיץ.
4. הוסיפו בקישור תמונה של טלפון קטן.
5. מקמו את התמונה בצמוד לצד הימני של המסך.
6. קבעו שהתמונה תהיה קבוע גם בעת גלילת הדף.

תרגיל 74

1. פתחו את המסמך שיצרתם בתרגיל 72.
2. הוסיפו את המילים כל הזכויות שמורות, בתוך פסקה בתוך התגית footer.
3. קבעו שהה footer ייצמד לתחתית הדף, גם במקרה של גלילה.
4. קבעו ל' footer צבע רקע כרצונכם.
5. קבעו לטקסט צבע כרצונכם, מקמו אותו במרכז.
6. קבעו ל- footer מאפיין רוחב: 100% (width:100%).

התוצאה תראה כך:



4. נסו להוסיף תוכן לדף בכדי לוודא שאכן ה footer נשאר צמוד בגלילה.

פתרון ה-css להגדרת ה-footer:

```
footer {font: 25px arial; position: fixed; bottom:0; background-color:black; color:white; width:100%; text-align:center;}
```

floating הצפת אלמנטים ו-z-index

z-index

ראינו שכשאנחנו ממקמים אלמנטים בדף ומגדירים להם מיקום position אבסולוטי או יחסי, לעיתים מספר אובייקטים יכולים לעלות אחד על גבי השני, לפעמים יתכן שנרצה לשנות את סדרם בסדר שונה מזה שנוצר, כשאחד יעלה על גבי השני, ניתן לקבוע את מיקום האובייקט מבחינת רמתו השכבתי באמצעות התכונה z-index, תכונה זו מקבלת מספרים, המספרים יכולים להיות מספרים חיוביים או שליליים.

כשנרצה להרים אובייקט על גבי אובייקט אחר, נעלה לו את רמת ה-z-index, במידה ונראה שעדיין האלמנט לא עולה על גבי האלמנט השני, ננסה להרים מעט יותר את ערכו של ה-z-index או לחילופין ניתן לאלמנט השני z-index שלילי, בכדי שוודאי ירד תחת האלמנט שאנו מעוניינים שיעמוד בקדמה.

לדוגמא הקוד הבא:

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div class="par1">פסקה אחת</div>
  <div class="par2">עוד פסקה</div>
</body>
</html>
```

הגדרנו לו את מאפייני ה-CSS הבאים:

```
.par1 { background-color: blue; width: 250px; height: 80px; position:
absolute; top: 80px; left: 100px;}
```

```
.par2 { background-color: red; width: 250px; height: 80px; position: absolute; top: 50px; left: 150px; }
```

בתצוגה בדפדפן זה נראה כך:



אם נרצה לשנות את סדר האלמנטים כך שהפסקה האדומה תופיע תחת הכחולה, נוסיף לפסקה הכחולה את התכונה `z-index`, כשניתן לו את הערך 2 והפסקאות תשגנה את סדרן כך:

```
z-index:2;
```

וכך זה נראה בדפדפן:



Floating - הצפת אלמנט

אחת הדרכים למיקום אלמנטים זה לצד זה, היא באמצעות התכונה: `float`, משתמשים בתכונה זו כאשר רוצים לדוגמא למקם תמונה מימין או משמאל לפסקה.

התכונה `float` מקבלת את אחד מהערכים הבאים:

`right` - להצפת האלמנט ימינה.

`left` - להצפת האלמנט שמאלה.

`none` - למניעת וביטול הצפה של אלמנט.

לדוגמא הקוד הבא:

```
<body dir="rtl">
```

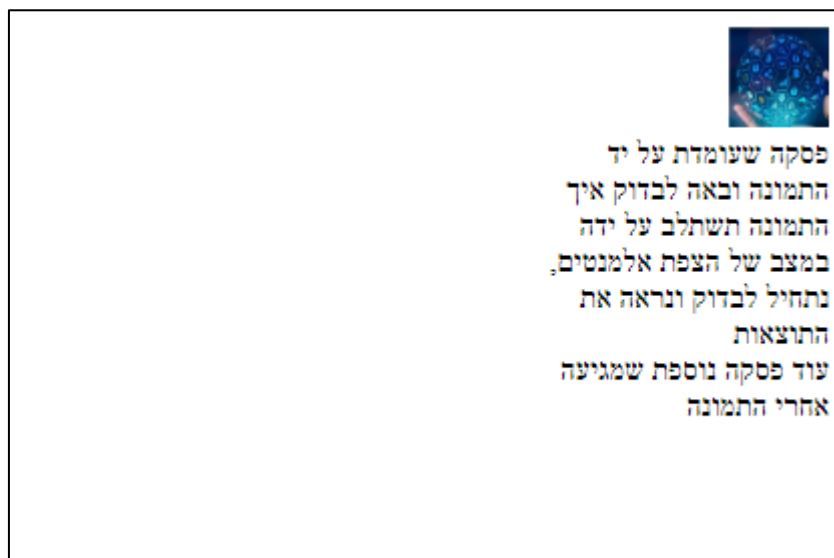
```

```

<div class="par1"> פסקה שעומדת על יד התמונה ובאה לבדוק איך התמונה תשתלב </div>
</div> על ידה במצב של הצפת אלמנטים, נתחיל לבדוק ונראה את התוצאות

<div class="par2"> עוד פסקה נוספת שמגיעה אחרי התמונה </div>
</body>

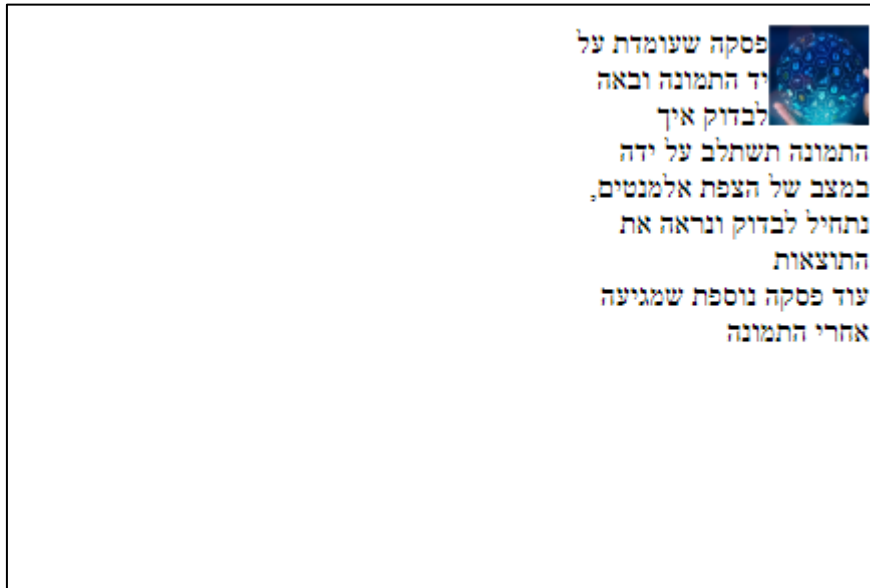
כך הוא נראה בדפדפן, מוצגת תמונה ותחתיה טקסט:



כאשר נגדיר לתמונה את המאפיין

Img {float:right; }

כך זה יראה, התמונה תצוף מימין לאובייקט שבה אחריה (הטקסט שמונח ב-div):



אם נקבע את המאפיין left

`Img {float:left; }`

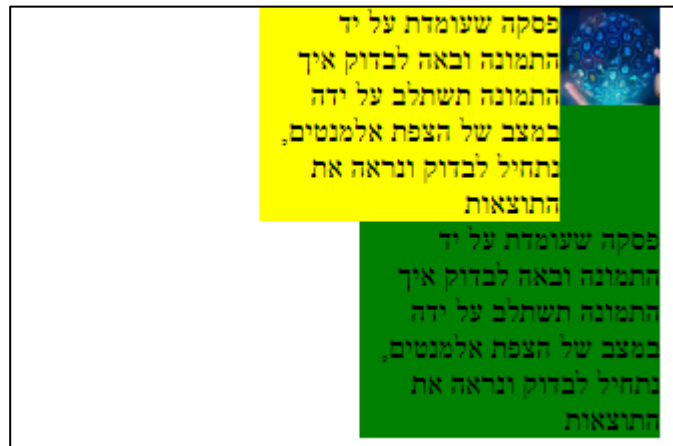
כך זה יראה, התמונה תצוף לצד שמאל:



התכונה clear

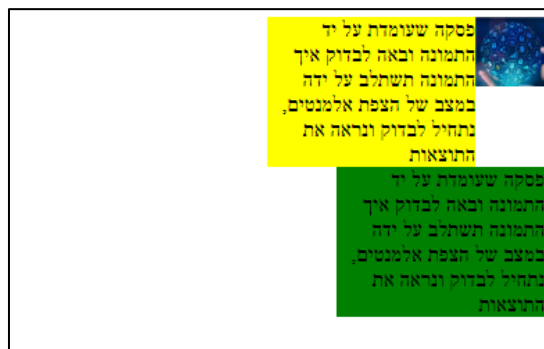
התכונה clear משמשת לביטול ציפה סביב אלמנט, וקובעת שהתוכן שיבוא ימוקם בשורה חדשה, ולא ימשך עם האלמנטים שהוגדרה להם ציפה.

לדוגמא:



זוהי תצוגה של דף בו שתי פסקאות ותמונה, לתמונה ולפסקה בעלת הרקע הצהוב, מוגדרת ציפה ימינה, הפסקה השלישית נמשכת אחריהם, על מנת למנוע את משיכתה של הפסקה השנייה, שפסקה זו תעמוד בשורה נפרדת חדשה, יש להגדיר את המאפיין clear עבור הפסקה השנייה, כשהערך שהיא תקבל הוא right, על מנת לבטל את הציפה מימין.

כך:



`.par2 {clear:right;}`

תכונה זו מקבלת את אחד משלושת הערכים הבאים:

right - לניקוי השפעת ה-float מימין, לביטול ציפה מימין.

left - לניקוי השפעת ה-float משמאל, לביטול ציפה משמאל.

both - לניקוי השפעת ה-float משני הכיוונים, לביטול ציפה משני הכיוונים יחד.

none - ללא ניקוי השפעת ה-float.

1. פתח מסמך html חדש,
2. הוסף במסמך 6 תמונות.
3. קבעו לתמונות גובה של 70px ורוחב של 150px.
4. קבעו לשלושת התמונות הראשונות ציפה ימינה,
5. לשתי התמונות הבאות קבעו ציפה שמאלה.
6. לתמונה השישית קבעו שלא תמשך אחר שאר התמונות, ותעמוד בשורה חדשה.

תוצאה:

```
<!DOCTYPE html>

<html>

<head>

  <title>עיצוב דף</title>

  <style>

    img{width: 150px; height:70px;}

    .right{float:right; }

    .left{float:left; }

    .clear{ clear:both;}

  </style>

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

</head>

<body dir="rtl">

  
```

```
  
  
  
  
<div class="clear"></div>  
  
</body>  
</html>
```

תרגיל 76

1. צרו מסמך html חדש.
2. הוסיפו כותרת "כלי נגינה".
3. הוסיפו כיתוב על שלושה כלי נגינה.
4. לצד כל כיתוב מקמו תמונה מתאימות.
5. קבעו שהטקסט יגלוש סביב לתמונה.

Pseudo Classes - קלאסים מדומים, דינאמיות של קישורים

Pseudo Classe - קלאסים מדומים

למדנו על סלקטורים- בוררים, באמצעות הסלקטור אנו מתארים את האובייקט או האובייקטים עליהם יחול העיצוב. (על ידי שם מחלקה, מזהה ייחודי או שם אלמנט) לעיתים נרצה להחיל עיצובים מסוימים לאלמנט במצב מסוים לדוגמה עיצוב לקישור כשעוברים עליו, נגדיר באמצעות זאת באמצעות קלאס מדומה (pseudo class).

המבנה התחבירי של קלאס מדומה

הקלאס המדומה, הוא קלאס שיתלווה לסלקטור מסוים, כשבין הסלקטור לקלאס יופיע הסימן נקודתיים (:), הקלאסים המדומים הם מתארים מצב מסוים של הסלקטור כך:

```
{;ערך: מאפיין} pseudo-class: סלקטור
```

לדוגמא:

```
p:hover{ color:red; }
```

דוגמא זו מגדירה שכשעוברים על הפסקאות במסמך, הטקסט שלהן יהפוך לאדום.

עיצוב קישורים באמצעות קלאסים מדומים

ניתן לתאר את ארבעת המצבים של קישור באמצעות קלאסים מדומים המתארים את מצבי הקישור:

1. **link** - מתאר את הקישור במצב שעדין לא בקרו בו.

לדוגמא:

```
a:link {color: black;}
```

2. **hover** - מתאר את הקישור במצב שעוברים עליו עם העכבר.

לדוגמא:

```
a:link {color: green;}
```

3. **active** - מתאר את הקישור במצב פעיל, מתאר את הקישור בזמן שהוא פעיל, בזמן הלחיצה (את משמעותו נראה כשהלחיצה תהיה ארוכה).

לדוגמא:

```
a:link {color: orange;}
```

4. **visited** - מתאר את הקישור במצב שביקרו בו. (כבר לחצו עליו).

לדוגמא:

```
a:link {color: red;}
```

ניתן לתת את הקלאסים המדומים גם על קישורים שממוקמים בהיררכיה ספציפית, או בעלי קלאס מסוים, העיקרון הוא שנוסיף את הסימן נקודתיים ושם הקלאס המדומה.

לדוגמא נרצה להגדיר לטקסט בקישור במצב שעוברים עליו שיקבל צבע אדום, הקישור ממוקם בתוך div שיש לו קלאס example נעשה זאת כך:

```
div.example a:hover { color:red; }
```

focus - לא קשור דווקא לקישורים, אך גם קלאס מדומה המתאר מצב דינאמי, תפקידו לתאר שמהו נמצא בפוקוס ומחכה שיזינו מידע לתוכו, לדוגמא שדה בטופס שנמצא בפוקוס, הפוקוס משמש לסמן על איזה אלמנט אנחנו עומדים כרגע, ובכך מסייע לנגישות לכאלה המוגבלים לשימוש עם העכבר, משתמשים דווקא במקלדת, ועוברים בין האלמנטים השונים באמצעות המקש tab.

לדוגמא:

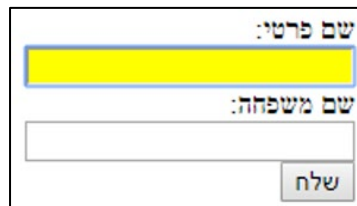
```
Input:focus, textarea:focus {
```

```
    Background: yellow
```

```
}
```

תרגיל 77

1. בנו טופס, הטופס יכיל:
2 תיבות טקסט, אחת להזנת שם פרטי, אחת להזנת שם משפחה
כפתור שליחה.
2. הגדירו שתיבה שהגולש בפוקוס עליה צבע הרקע שלה נעשה צהוב.
לדוגמא, כך תראה התיבה שם פרטי, כאשר המשתמש נמצא עליה בפוקוס:



תרגיל 78

1. הכינו תפריט ובו קישורים ל4 דפים: אודות, דף הבית, צור קשר, מתוצרתנו.
2. הקישורים כרגע לא יקושרו לדף כלשהו, אלא יוגדרו כריקים (`href="#"`).
3. הקישורים יעוצבו בעיצוב הבא: צבע רקע - כחול, מסגרת בעובי של 2 פיקסלים בצבע שחור, גודל הקישורים - רוחב: 50 פיקסלים, גובה 40 פיקסלים, צבע טקסט לבן.
4. בעת מעבר על אחד מהקישורים, הקישור יקבל צבע רקע, כחול בהיר, צבע טקסט שחור.

תרגיל 79

1. הכינו מסמך ובו 3 div, בכל אחד מילוי רקע של תמונה אחרת.
2. בעת מעבר עכבר על כל אחד מהם תשתנה תמונת הרקע לתמונת רקע שונה.

תרגיל 80

העתיקו את קוד ה-html הבא למסמך html חדש, מקמו את הקוד בתגית ה-body.

```
<p><a class="one" href="#" target="_blank">קישור ראשון</a></p>
```

```
<p><a class="two" href="#" target="_blank">קישור שני</a></p>
```

```
<p><a class="three" href="#" target="_blank">קישור שלישי</a></p>
```

<p>קישור רביעי</p>

<p>קישור חמישי</p>

a. הגדירו לקישור הראשון צבע כתום במעבר עכבר.

b. הגדירו לקישור השני שיגדל פי 1.5 בעת מעבר עכבר (גודל גופן 150%).

2. הגדירו לקישור השלישי צבע רקע ירוק בעת מעבר עכבר.

3. הגדירו לקישור הרביעי שהקו התחתני שלו יעלם בעת מעבר עכבר.

4. הגדירו לקישור החמישי, שיקטן לגודל 50% בעת מעבר עכבר.

פתרון:

```
a.one:hover {color: orange;}
```

```
a.two:hover {font-size:150%;}
```

```
a.three:hover {background:green;}
```

```
a.four:hover {text-decoration:none;}
```

```
a.five:hover {font-size:50%;}
```

עיצוב סמן העכבר (cursor)

באמצעות מאפיין CSS זה ניתן להגדיר את עיצוב סמן העכבר לעיצובים שונים.

הערכים שניתן לתת למאפיין זה הם רבים נציג כמה מהם:

צורה של סימן שאלה:

```
a { cursor: help; }
```

צורה של יד:

```
a { cursor: pointer; }
```

צורה של אסור:

```
a { cursor: no-drop; }
```

תוכלו לצפות ברשימה נרחבת בקישור הבא:

https://www.w3schools.com/cssref/tryit.asp?filename=trycss_cursor

תרגיל 81

הגדירו בתרגיל 2 סמן בצורת יד, בעת מעבר עכבר על קישורי התפריט.

pseudo elements - אלמנט מדומה

ראינו ש-pasoudo class מטפל במצב מסוים של האלמנט. בנוסף, קיימת אפשרות להתייחס לאלמנט לפי הגדרה מסוימת לדוגמא האובייקט הראשון במסמך, השורה הראשונה, האות הראשונה וכדומה. כאשר הגדרנו שהפסקה הראשונה מבין קבוצת פסקאות תקבל עיצוב מסוים, כשנשנה את המסמך ונוסיף פסקה חדשה לפני זו שהייתה ראשונה, הפסקה שהוספנו היא זו שתענה להגדרה, כלומר העיצוב יחול על האלמנט שיענה להגדרה שתוגדר

לדוגמא:

```
P:first-child { color:red;}
```

דוגמא זו מגדירה לפסקה הראשונה בדף שתקבל עיצוב אדום, כלומר מוסיפה לסלקטור פסקה, התייחסות לאלמנט שעומד הראשון בדף.

מספר דוגמאות לאלמנטים מדומים:

first-child - בן ראשון, האלמנט הראשון.

last-child - בן אחרון, האלמנט האחרון.

שתי דוגמאות:

1. `p span:first-child { color: blue; }`
2. `p:first-child span { color: blue; }`

מה ההבדל בין שתי ההגדרות הנ"ל?

ההגדרה הראשונה קובעת לכל `span` ראשון בתוך התגית `p` שיצבע בצבע כחול, כך שגם אם יש עשרות פסקאות ובכל אחת מהם מספר אבייקטי `span`, בכל פסקה רק הן `span` הראשון יקבל עיצוב כחול.

ההגדרה השנייה קובעת לכל `span` שהוא בתוך פסקה ראשונה בעמוד שיצבע בצבע כחול, כך שגם אם יש מספר אובייקטי `span` בתוך התגית `p` הראשונה בעמוד הן תצבענה בצבע כחול.

not(selector) - מתאר את כל האלמנטים שהם לא עונים על הגדרת סלקטור מסויים.

לדוגמא:

```
:not(p) { background: blue; }
```

הגדרה זו קובעת צבע רקע כחול לכל האלמנטים שאינם פסקה.

nth-child(number) - מתאר את האלמנט במיקום מספרי מסוים.

לדוגמא:

```
p:nth-child(2) { color: green; }
```

הגדרה זו קובעת צבע ירוק לטקסט של הפסקה השנייה בעמוד.

nth-last-child(number) - אותו דבר כמו nth-child(n) רק ספירת האלמנטים הפוכה, מתחילה מהאלמנט האחרון.

first-line - מעצב את השורה הראשונה באלמנט

לדוגמא:

```
p::first-line { color: red; }
```

בדוגמא זו השורה הראשונה בפסקה תעוצב בצבע אדום, נראה את משמעות הקוד בפסקה ארוכה המתפרסת על פני יותר משורה בודדת.

first-letter - מעצב את האות הראשונה בפסקה

לדוגמא:

```
p::first-letter {  
  color: red;  
  font-size: 2em;  
}
```

בדוגמא זו האות הראשונה בפסקה תעוצב בצבע אדום, וגדלה יוגדר פי 2 מגודלן של שאר האותיות בפסקה.

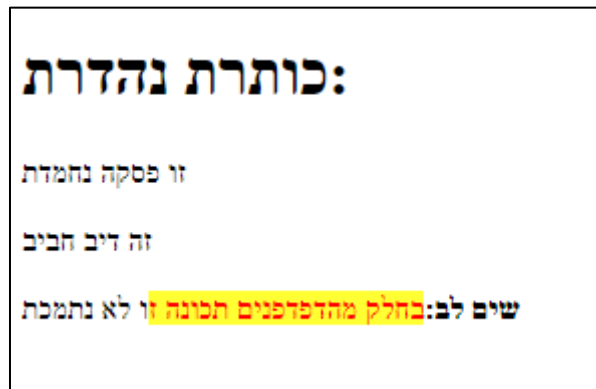
::selection - קובע את עיצוב האלמנט כשהגולש מסמן אותו (כשהוא לוחץ על העכבר וגורר אותו על פני האלמנט באתר, כמו סימון טקסט או כל אלמנט שהוא).

לדוגמא:

```
::selection { color: red; background: yellow; }
```

בדוגמא זו קבענו שכשהגולש יסמן אלמנט באתר, האלמנט יקבל צבע טקסט אדום, וצבע רקע צהוב.

דוגמא לתצוגה כשהגולש מסמן את התווים "בחלק מהדפדפנים תכונה זו":



שים לב: יתכן שתכונה זו לא נתמכת על ידי חלק מהדפדפנים.

תרגיל 82

1. הכן פסקה ארוכה שתכיל טקסט המתפרס על מספר שורות. (ניתן למלא עם טקסט אקראי כמו לורם איפסום וכדו').
2. שכפל את הפסקה מספר פעמים.
3. קבע שצבע הטקסט בשורה הראשונה בכל פסקה יהיה בצבע כתום.
4. קבע שהאות הראשונה בכל פסקה תגדל פי 4 (4em).
5. הוסף רשימה של 3 פריטים.
6. שנה את סימן הרשימה לסימן ~.
7. קבעו שכשהגולש מסמן תווים במסמך התווים המסומנים יקבלו צבע רקע כתום.

ב-CSS קיימים עוד מספר קלאסים מדומים, פרטנו את העיקריים שבהם.

ניתן למצוא את רשימות מפורטות יותר, להלן מספר קישורים לעיון:

https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Pseudo-classes_and_pseudo-elements

<https://www.quackit.com/css/selectors/>

https://www.w3schools.com/css/css_pseudo_classes.asp

before ו-after

ניתן להוסיף באמצעות CSS טקסטים, קישורים עיצוביים, או תמונות, הקישורים יופיעו לפני או אחרי אלמנט בדף, המאפיין content קובע את התוכן של הקישוט שיופיע לפני או אחרי האלמנט, התכונה **content** היא חובה, ולא ניתן לוותר עליה.

לדוגמא:

```
a:before{ content:"link"; }
```

דוגמא זו תוסיף לפני כל קישור את המילה: "link".

התכונה content מכניסה את התוכן לדף כאילו היא כתובה בתוך ה-html. כמובן לא מומלץ להכניס באופן זה טקסט חשוב, אלא אך ורק דברים השייכים לעיצוב.

ניתן גם להגדיר שהתוכן שיבוא לפני האלמנט יהיה תמונה לדוגמא:

```
h1::before { content: url(smiley.png); }
```

דוגמא זהו תוסיף את התמונה שהגדרנו לפני הכותרות h1.

למה זה שימושי?

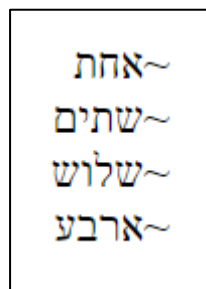
כמו שהרחבנו CSS לא בא להוסיף תוכן, אלא מטרתו היא עיצוב הדף, גם לתוספות שניתן להוסיף באמצעות ה-CSS באמצעות before ו-after נתייחס כאל טקסט עיצובי, ולא טקסט סתם.

משתמשים בזה לדוגמא אם רוצים לעצב רשימה ובמקום להשתמש באפשרויות התבליטים המוכרות כמו נקודה ריבוע וכדו', נוכל להשתמש באפשרויות נוספות על ידי ביטול תבליט

ברירת המחדל והוספת סימן באמצעות האלמנט המדומה before לפני כל פריט ברשימה
כך:

```
li{  
  list-style-type:none; //ביטול התבליט הבסיסי  
  position:relative;  
}  
li:before{  
  content: '~'; //הוספת קישוט במקום התבליט שביטלנו  
}
```

לאחר הגדרה זו הרשימה תראה כך:



תרגיל 83

1. פתחו מסמך html חדש.
2. צרו רשימה של פרטי לבוש.
3. הוסיפו את הסימן < לפני כל פריט.
4. היעזרו בסלקטור before.

תרגיל 84

1. פתחו מסמך html חדש.
2. צרו רשימה של פרטי לבוש.
3. שנו את התבליט לצד כל שורה לציור של גלובוס קטן.
4. היעזרו בסלקטור before.

תרגיל 85

צרו מסמך html חדש.

הוסיפו 5 פסקאות, מלאו אותן בתוכן.

לאחר כל פסקה הוסיפו את הטקסט "חדש".

היעזרו ב-after.

media query רספונסיביות ונגיעות בגריד של bootstrap**רספונסיביות**

Responsive בעברית זו תגובה, התאמה רספונסיבית זו התאמת מראה האתר האתר יוכל להגיב לשינויי גודל מסך ולהתאים את עצמו לכל מסך של כל גולש וגולש, בין אם הגולש גולש ממחשב שולחני בעל מסך רחב, ובין אם הגולש גולש ממכשיר הנייד, בין אם הגולש מציג את הדפדפן על כל רוחב המסך ובין אם הגולש מציג את הדפדפן רק על חלקו של המסך.

חוסר רספונסיביות

כאשר האתר לא מותאם לתצוגות מסך שונות, במסך צר, יתכן שנזדקק לגלול את הפס גלילה האופקי על מנת לצפות בתוכן, מה שמאד לא ידידותי למשתמש, או יתכן שחלק מהטקסטים יחולקו לעמודות מידי צרות, כך שיהיה קשה לקרוא אותם מתוך מסכים צרים.

איך עושים את ההתאמה?**Media query**

השיטה היא לעטוף מספר פקודות, ככל שנרצה בmedia queries זהו תקן של css3 המאפשר להציג את אותו בלוק נתונים בצורה אחת, כשמתקיימים תנאים מסוימים, ובצורה שונה כשמתקיימים תנאים אחרים, כך שהוא מאפשר להגדיר הגדרות נפרדות לאותם בלוקים בתנאים שונים, media query מתייחס לסוג המסך (נייט, טאבלט או נייד), לגודלו, למנח המסך (מנח אופקי או אנכי), רזולוציית המסך ועוד.

מבנה ה-media query

```
@media mediatype and (mediafeature and|or|not mediafeature){
```

```
    הגדרות CSS
```

```
}
```

ערכים מקובלים למאפיין media type:

all - כל סוגי המדיה הקיימים.

print - שימושי להדפסה, אם נרצה להגדיר הגדרות מסוימות שתתקבלנה כשמדפיסים את הדף, לדוגמא אם נרצה להגדיר שהטקסטים בתצוגת מסך יקבלו צבע ירוק, ואילו שכשמדפיסים את הדף, כל הטקסטים יקבלו צבע שחור, נעשה זאת כך:

```
@media screen {  
  body {  
    color: green;  
  }  
}
```

```
@media print {  
  body {  
    color: black;  
  }  
}
```

screen - שימושי למסכי מחשב, טבלאטים, סמארטפונים וכדו'.

speech - שימושי עבור קוראי מסכים.

ערכים שימושיים למאפיין *mediafeature*

בערך זה נגדיר את התנאים שאנו מפעילים בפקודת ה-*media query*, ניתן להתייחס למגוון נושאים כמו גודל מסך, רזולוצית מסך, כמות צבעים ועוד.

הערכים השימושיים מאד מבין הערכים האפשריים הם:

min-width - רוחב מינימום, תנאי זה מגדיר רוחב מינימום למסך, מסכים העונים להגדרת הרוחב שמוגדר, הם מסכים בעלי הגודל המוגדר ומעלה, מסכים אלה יקבלו את התכונות שיוגדרו בתוך ה-*media query*.

max-whidth - רוחב מקסימום, תנאי זה מגדיר רוחב מקסימאלי למסך, מסכים העונים להגדרת הרוחב שמוגדר, הם מסכים בעלי הרוחב המוגדר ומטה, מסכים אלה יקבלו את התכונות שיוגדרו בתוך ה-media query.

שיטת עבודה מקובלת היא דבר ראשון לפתח את קוד ה-html בהתאמה למסכים רחבים, ולאחר מכן להגדיר נקודות שבירה כשרוחב המסך גדול מנקודת השבירה שהגדרנו, יוצג הקוד הראשוני שפתחנו, מסכים שהם קטנים מנקודת השבירה, יקבלו את שינויי ה-css שהגדרנו בתוך ה-media query שמציינת את נקודת השבירה.

לדוגמא לפנינו מסך המכיל 4 עמודות, העמודות זו לצד זו, העמודות מכילות טקסט, הגדרנו לכל אחת מהן רוחב של 22% מרוחב המסך, המסך נראה כך:

וידאו	וידאו	וידאו	וידאו
<p>וידאו מספר דרוי ריבה-ישעבה טקסטורי דר הויכוח את תמונה טור. בעת הליכה על וידאו מספר, באנטייפיק הויכוח את הסדר הטובב על הוויכוח טקסטורי הוויכוח. ניתן גם להחליף טיול טמטת מי הויכוח בויכוח מספר את ויכוח הויכוח פוהר הויכוח הויכוח טור. בני הויכוח הויכוח טור ויכוח הויכוח, מספר שויכוח על הויכוח הויכוח טור. הויכוח הויכוח הויכוח טור. הויכוח הויכוח הויכוח טור.</p>	<p>וידאו מספר דרוי ריבה-ישעבה טקסטורי דר הויכוח את תמונה טור. בעת הליכה על וידאו מספר, באנטייפיק הויכוח את הסדר הטובב על הוויכוח טקסטורי הוויכוח. ניתן גם להחליף טיול טמטת מי הויכוח בויכוח מספר את ויכוח הויכוח פוהר הויכוח הויכוח טור. בני הויכוח הויכוח טור ויכוח הויכוח, מספר שויכוח על הויכוח הויכוח טור. הויכוח הויכוח הויכוח טור. הויכוח הויכוח הויכוח טור.</p>	<p>וידאו מספר דרוי ריבה-ישעבה טקסטורי דר הויכוח את תמונה טור. בעת הליכה על וידאו מספר, באנטייפיק הויכוח את הסדר הטובב על הוויכוח טקסטורי הוויכוח. ניתן גם להחליף טיול טמטת מי הויכוח בויכוח מספר את ויכוח הויכוח פוהר הויכוח הויכוח טור. בני הויכוח הויכוח טור ויכוח הויכוח, מספר שויכוח על הויכוח הויכוח טור. הויכוח הויכוח הויכוח טור. הויכוח הויכוח הויכוח טור.</p>	<p>וידאו מספר דרוי ריבה-ישעבה טקסטורי דר הויכוח את תמונה טור. בעת הליכה על וידאו מספר, באנטייפיק הויכוח את הסדר הטובב על הוויכוח טקסטורי הוויכוח. ניתן גם להחליף טיול טמטת מי הויכוח בויכוח מספר את ויכוח הויכוח פוהר הויכוח הויכוח טור. בני הויכוח הויכוח טור ויכוח הויכוח, מספר שויכוח על הויכוח הויכוח טור. הויכוח הויכוח הויכוח טור. הויכוח הויכוח הויכוח טור.</p>

כל הכותיות שמורות

כאשר נקטין את רוחב המסך, העמודות תוצרנה, לפי רוחב המסך החדש, כשהמסך יהיה ממש צר העמודות לא תראינה טוב, השורות תראינה מידי קצרות כך:

וידאו	וידאו	וידאו
וידאו מספק דרך רבת-עוצמת שתעזור לך להוכיח את הנקודה שלך. בעת החליצה על "וידאו מקוון", באפשרותך להרביץ את הקוד המוטבע של הווידאו שברצונך להוסיף. ניתן גם להקמיד מילת מפתח כדי להפיש באופן מקוון אחרי וידאו שיתאים בצורה הטובה ביותר למסךך. כדי להעניק למסךך שלד מראה משניוני, Word מספק עיצובים של כותרת עליונה, כותרת תחתונה, עמוד שער ותוכנית טקסט אשר משלימים זה את זה. לדוגמה, באפשרותך להוסיף עמוד שער, כותרת עליונה ומרחב צדדי ותאמים.	וידאו מספק דרך רבת-עוצמת שתעזור לך להוכיח את הנקודה שלך. בעת החליצה על "וידאו מקוון", באפשרותך להרביץ את הקוד המוטבע של הווידאו שברצונך להוסיף. ניתן גם להקמיד מילת מפתח כדי להפיש באופן מקוון אחרי וידאו שיתאים בצורה הטובה ביותר למסךך. כדי להעניק למסךך שלד מראה משניוני, Word מספק עיצובים של כותרת עליונה, כותרת תחתונה, עמוד שער ותוכנית טקסט אשר משלימים זה את זה. לדוגמה, באפשרותך להוסיף עמוד שער, כותרת עליונה ומרחב צדדי ותאמים.	וידאו מספק דרך רבת-עוצמת שתעזור לך להוכיח את הנקודה שלך. בעת החליצה על "וידאו מקוון", באפשרותך להרביץ את הקוד המוטבע של הווידאו שברצונך להוסיף. ניתן גם להקמיד מילת מפתח כדי להפיש באופן מקוון אחרי וידאו שיתאים בצורה הטובה ביותר למסךך. כדי להעניק למסךך שלד מראה משניוני, Word מספק עיצובים של כותרת עליונה, כותרת תחתונה, עמוד שער ותוכנית טקסט אשר משלימים זה את זה. לדוגמה, באפשרותך להוסיף עמוד שער, כותרת עליונה ומרחב צדדי ותאמים.
וידאו		
וידאו מספק דרך רבת-עוצמת שתעזור לך להוכיח את הנקודה שלך. בעת החליצה על "וידאו מקוון", באפשרותך להרביץ את הקוד המוטבע של הווידאו שברצונך להוסיף.		
כל הזכויות שמורות		

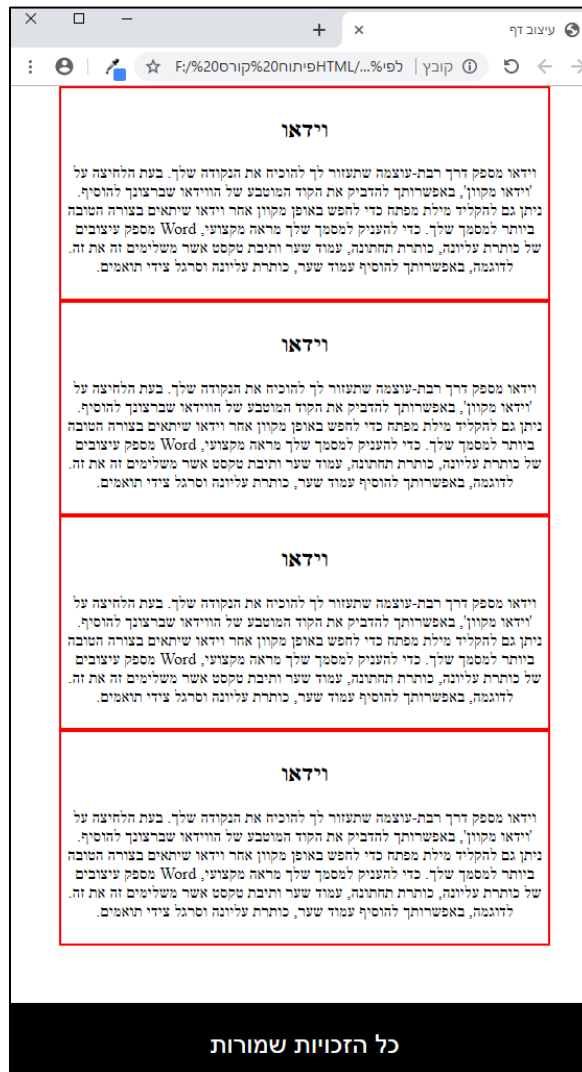
בדוגמא כזו נרצה להגדיר שבמסך צר, העמודות תקבלנה את מלא רוחב העמודה, והן תהינה ברוחב של קרוב ל-100%.

נתן הגדרת שבירת מסך, שכל המסכים שגדלם קטן מ-768, העמודות תקבלנה בהם רוחב מלא של 80%.

ההגדרה שהוספנו:

```
@media screen and (max-width: 768px) {
    div{width:80%; }
}
```

קעת כאשר האתר יוצג במסך צר התצוגה תראה כך:



ניתן גם להשתמש ב-media query בכמה תנאים בו זמנית לדוגמא:

```
@media screen and (max-width: 900px) and (min-width: 600px), (min-width: 1100px) {
```

CSS פקודות

```
}
```

Breakpoints - הגדרת נקודת השבירה ב-CSS

מספר נקודות שבירה חשובות:

אם נרצה להתאים את המסך באופן שונה לכל אחד מסוגי המכשירים הקיימים, נצטרך להגדיר את נקודות השבירה הבאות:

768 - מסכים של מכשירי טלפון ניידים בד"כ בגודל המקסימלי שלהם הוא 768 פיקסלים.

768-992 - טאבלטים הם בד"כ בגודל שבין 768 פיקסלים ל-992 פיקסלים.

992-1200 - מסכי מחשב הם בד"כ בגודל שבין 992 פיקסלים עד 1200 פיקסלים.

מעל 1200 - מסכי מחשב רחבים יותר הם מעל 1200 פיקסלים.

את נקודות השבירה נגדיר באמצעות `media query`, את ה-`media query` נגדיר בסוף קוד ה-CSS, ומכיוון שמיקום ההגדרות הוא במקום מאוחר בסדר פקודות ה-CSS במידה והמסך עונה על תנאי ההגדרה ב-`media query` הוא יקבל את הגדרות ה-CSS שמוגדרות בו.

לדוגמא נגדיר `media query` למסכים עד גודל של 768 כך:

```
@media screen and (max-width: 768px) {
```

```
  כאן תכתבנה פקודות ה-CSS שתקבלנה שינוי
```

```
}
```

נוכל להגדיר מספר פקודות `media query` לנקודת שבירה שונות, כשבכל אחת השינויים הנזקקים לפי גודל המסך.

הצגה או הסתרה של אלמנטים

פעמים רבות יהיו אלמנטים שנרצה להציג או להסתיר בתצוגות מסך שונות, בכדי להסתיר או להציג אלמנטים שונים, נשתמש בפקודה `display:none` להסתרת אלמנטים, ובפקודה `display: block` בכדי להציג אלמנטים שהסתרו. (נושא ה-`display` הורחב באחד מהשיעורים הקודמים)

1. צרו מסמך html חדש.
2. הוסיפו למסמך תמונה.
3. קבעו לתמונה ברוחב של 100 פיקסלים וגובה של 50 פיקסלים.
4. הוסיפו תמונה נוספת, הגדירו לתמונה רוחב של 1000 פיקסלים.
5. הוסיפו תחת התמונה הגדולה div המכיל את הטקסט "תמונה נהדרת".
6. עטפו את התמונה הגדולה וה-div עם הטקסט בתוך div נוסף.
7. עטפו את התמונה הקטנה, וה-div של התמונה הגדולה, בתוך div.
8. קבעו שהתמונה הגדולה והטקסט תחתיה יוסתרו (display: block).
9. בעת מעבר על התמונה הקטנה התמונה הגדולה תוצג.
10. קבעו שבמסך שגודלו קטן מ768 פיקסלים, רוחב התמונה הגדולה יהיה 300 פיקסלים.

תשובה:

```
<!DOCTYPE html>
<html>
<head>
<style>
    .dropdown { position: relative; display: inline-block;}
    .dropdown-content { display: none; position: absolute;
background-color: #f9f9f9; min-width: 160px; z-index: 1;}
    .dropdown:hover .dropdown-content { display: block;}
    .desc { padding: 15px; text-align: center;}
    #bigimg{width:1000px;}
    @media screen and (max-width: 768px) {
        #bigimg{width:300px;}
    }
</style>
```

```

</head>
<body>
<div class="dropdown">
  
  <div class="dropdown-content">
    
    <div class="desc">תמונה נהדרת</div>
  </div>
</div>
</body>
</html>

```

תרגיל 87

1. צרו תפריט נפתח.
2. בדפדפן יוצג כפתור אחד מתוך התפריט.
3. בעת מעבר על הכפתור, יפתחו תחתיו 3 כפתורים נוספים.
4. בעת מעבר העכבר על הכפתורים הכפתורים ישנו את צבעם.
5. בעת מעבר העכבר סמן העכבר ישנו את צורתו לצורת יד.
6. קבעו שבמסך שגודלו קטן מ768 פיקסלים, התפריט יעמוד פתוח כל הזמן ברוחב של 100%.

```

<!DOCTYPE html>
<html>
<head>
<style >
.dropbtn { background-color: #4CAF50; color: white; padding: 16px;
font-size: 16px; border: none; cursor: pointer;}
.dropdown { position: relative; display: inline-block;}
.dropdown-content { display: none; position: absolute; background-
color: #f9f9f9;
  min-width: 160px;}
.dropdown-content a { color: black; padding: 12px 16px; text-
decoration: none;
  display: block;}
.dropdown-content a:hover {background-color: #f1f1f1}
.dropdown:hover .dropdown-content { display: block;}
.dropdown:hover .dropbtn { background-color: #3e8e41;}
@media screen and (max-width: 768px) { .dropdown .dropdown-
content { display: block; }
  button,a{width: 100%;}
}
</style>
</head>
<body dir="rtl">
<div class="dropdown">
  <button class="dropbtn">קישור ראשי</button>
  <div class="dropdown-content">
    <a href="#">קישור 1</a>

```

```
<a href="#">2 קישור</a>
<a href="#">3 קישור</a>
</div>
</div>
</body>
</html>
```

המאפיין `max-width`

מאפיין זה עוזר בהתאמת האתר לפלטפורמות השונות, מאפיין זה קובע רוחב מקסימאלי לאלמנט, כך שהאלמנט לא יגדל יותר מהרוחב שקבענו, אך במידת הצורך, כאשר לאלמנט לא יהיה מספיק מקום, הוא יקטן בהתאם לצורך, כאשר עובדים בהגדרת גדלים באחוזים הגדרת רוחב מקסימלי לתמונות היא כמעט הכרחית כי לעולם לא נרצה שהתמונה שלנו תמתח מעבר לגדלה המקורי, על מנת שלא תאבד מחיוניותה על ידי עיוות או פיקסול של התמונה כתוצאה מהגדלת יתר.

לדוגמא:

```
p.ex1 {
  max-width: 150px;
}
```

דוגמא נוספת:

```
img {
  max-width: 100%;
  height: auto;
}
```

ניתן גם לקבוע רוחב מינימלי, שהאלמנט לא יקטן ממנו באמצעות הפקודה `min-width`.

הגדרת גדלים באחוזים

את גדלי האלמנטים השונים מומלץ להגדיר באחוזים, כך שבכל מסך הגודל יותאם ללא צורך בהגדרת גדלים שונים לנקודות שבירה קרובות.

תרגיל 88

1. בנו דף, עם 4 אזורי טקסט (הניחו טקסט כרצונכם, טקסט חופשי, או טקסט מאתר כלשהו).
2. בכל אזור טקסט הוסיפו תמונה.
3. קבעו שרוחב כל עמודה יהיה 22%.
4. קבעו שכאשר המסך צר מ-768 פיקסלים, העמודות תקבלנה רוחב של 80%.

תרגיל 89

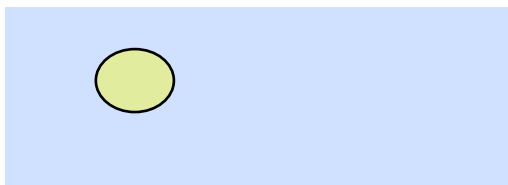
1. פתחו מסמך html חדש
2. הוסיפו במסמך 5 תמונות
3. 3 פסקאות ארוכות
4. קבעו שהתמונות תעמודנה בשורה
5. קבעו לפסקאות מסגרת
6. קבעו ששלושת הפסקאות תעמודנה זו לצד זו
7. כאשר המסך צר מ-768 פיקסלים, כל האלמנטים יקבלו רוחב של 70%.

איך נבדוק את האתר במסך צר?

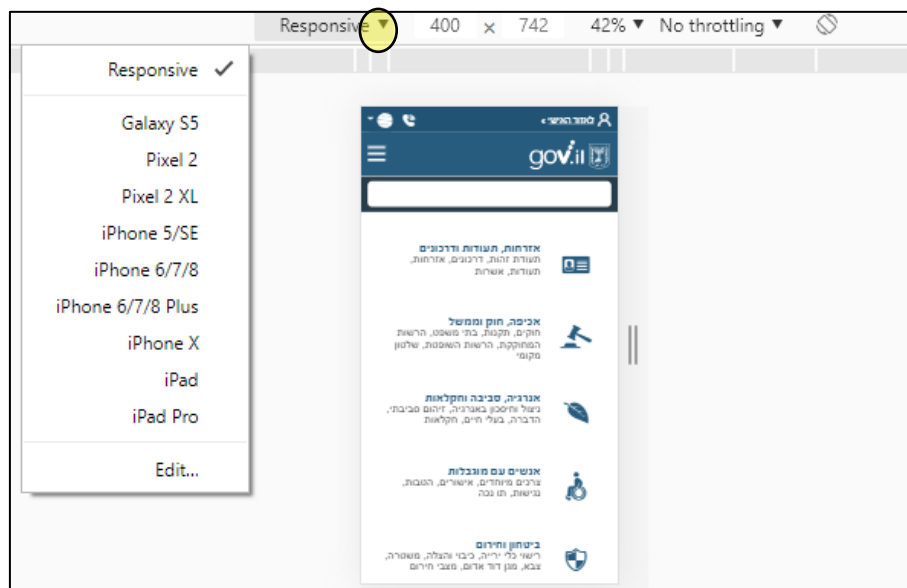
כמה דרכים לבדיקת נראות האתר במסך צר:

1. גלישה מטלפון חכם - אפשרות זו קיימת רק כאשר האתר יושב ברשת האינטרנט ולא במחשב המקומי.
2. ניתן להצר את חלון הדפדפן, כך שהדפדפן מקבל רוחב צר.
3. באמצעות ה"בדוק מרכיב" - f12.

כשניגש לאתר ונכנס למצב של בדוק מרכיב, על ידי אחד משני האופנים הבאים: לחיצה ימנית באתר ובחירה באפשרות: "בדוק", או לחיצה על מקש הקיצור f12, נמצא בצד השמאלי של חלונית בדיקת המרכיבים את הלחצן הבא:



בלחיצה על לחצן זה, נוכל לבדוק את נראות האתר בגלישה כיכול ממכשירים מסוגים שונים, את סוג המכשיר נבחר בחץ שלצד המאפיין responsive.



תרגיל 90

פתחו את התרגיל הקודם, בדקו את נראות הדף במכשירים השונים, באמצעות הבדוק מרכיב.

רספונסיביות על ידי שימוש ב-bootstrap

Bootstrap היא סביבת עבודה בקוד פתוח וחינמי, עוזרת למפתחי אתרים לעצב את נראות האתר שלהם בצורה אסטטית, ובין היתר גם ליצור אתר רספונסיבי המותאם לכל סוגי המכשירים.

סביבת הפיתוח של bootstrap כוללת מסמך css הכולל בתוכו אוסף ענק של מחלקות - קלאסים, קישור קובץ ה-html לקובץ ה-css של בוטסטרפא ושילוב הקלאסים של bootstrap באלמנטים במסמך ה-html גורמים לאלמנטים לקבל את הגדרות ה-css ש-bootstrap הגדירה, ההגדרות של bootstrap מאפשרות קבלת אתר רספונסיבי בקלות ללא מאמץ רק על ידי שתילת הקלאסים באלמנטים המתאימים.

בנוסף ל-bootstrap יש סקריפטי javascript רבים המאפשרים להוסיף תכונות מתקדמות לאתר, כמו חלונות קופצים, גרפים וכדומה.

ניתן ללמוד ולצפות בתכונות הקלאסים השונות של bootstrap בתאר שלהם:

<https://getbootstrap.com/>

על מנת להשתמש בקלאסים של bootstrap יש לשתול בתגית ה-head של המסמך את השורה הבאה:

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-
ggOyRoiXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2
MZw1T" crossorigin="anonymous">
```

ניתן בכל עת לגלוש לאתר של bootstrap על מנת להעתיק את הקישור לקובץ ה-css שלהם, כמו כן bootstrap מאפשרת להוריד את קובץ הגדרות ה-css שלה, כך שניתן להניח אותו באתרכם ולהשתמש בו גם ללא קישור לאתרם. (דרך זו פחות ממומלצת כיוון שמכבידה על האתר).

על מנת לקבל פונקציונאליות מהגדרות ה-js של bootstrap יש צורך לשתול בתגית head את השורות הבאות:

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965DzOorT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6
jizo" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-
UOzeToCpHqdSjQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dl
HNDzoW1" crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-
```

JjSmVgydop3pXB1rRibZUAYoIly6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07j
RM" crossorigin="anonymous"></script>

כמובן ניתן להעתיקם מהאתר של bootstrap.

שינוי צורה ופעולת השתנות

התכונה transform - שינוי צורה

התכונה transform מאפשר לשנות גרפיים לתכונות של אלמנטים, לשנות את הצורה, להגדיל, להקטין ולעוות אותה.

התכונה transform מקבלת את אחד מהערכים הבאים:

scale - שינוי מידות, הגדלה או הקטנה.

rotate - סיבוב, הטיית האלמנט.

skew - הטיה, מטה את האלמנט, מלכסן אותו, גורם למלבן להפוך למקבילית.

translate - משנה את מיקום האלמנט במסך, מזיז אותו למקום חדש.

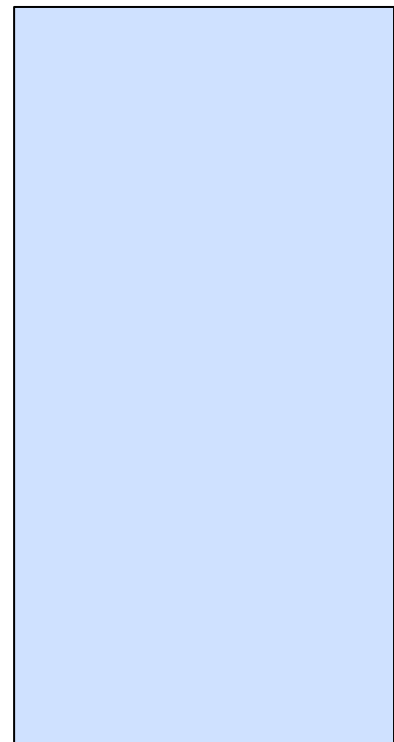
הערך **scale** משנה את מידת האלמנט לדוגמא:

```
<!DOCTYPE html>
```

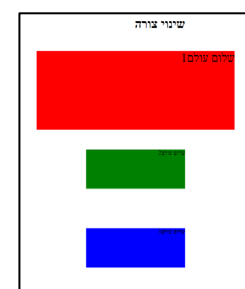
```

<html>
<head>
  <title>טופס</title>
  *{margin: 100px;}
  #one{ background:red;}
  #two{ background:green;}
  #three{ background:blue;}
  p{ width: 250px; height: 100px}
</style>
</head>
<body dir="rtl">
<h1>שינוי צורה</h1>
<p id="one">עולם 1</p>
<p id="two">עולם 2</p>
<p id="three">עולם 3</p>
</body>
</html>

```



אם נוסיף את המאפיין `transform: scale(2);` ל-`#one id`, הפסקה הראשונה תגדל פי שתיים ותראה כך:



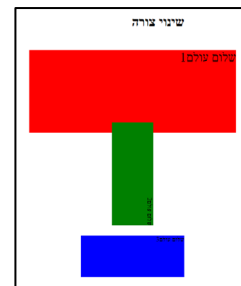
כאשר נעביר ל-`scale` שני ערכים נוכל לשנות את גודלו של האלמנט בציר ה-x בנפרד ובציר ה-y בנפרד, כלומר אורך האלמנט בנפרד ורוחב האלמנט בנפרד, הערך הראשון מציין את ציר ה-x - רוחב האלמנט, והערך השני מציין את ציר ה-y גובה האלמנט.

לדוגמא:

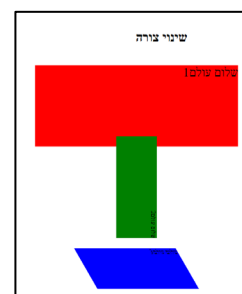
```
transform: scale(1.5, 0.5);
```

ערך זה יגרום לאלמנט לגדול ברוחב לפי אחד וחצי, ואילו ברוחב האלמנט יקטן לרוחב של חצי מרוחבו המקורי.

הערך rotate יגרום לסיבוב האלמנט אם נוסף את המאפיין `transform: rotate` ל-`#two id`; הפסקה השניה תסתובב בזווית של 90 מעלות ותראה כך:



הערך skew גורם להטיית האלמנט אם נוסף את המאפיין `transform: rotate` ל-`#three id`; הפסקה השלישית תקבל הטיה, עיוות במימד האופקי, ותראה כך:



הטיית ברירת מחדל היא בכיוון האופקי, ניתן גם להגדיר הטיה בכיוון האנכי באמצעות המאפיין `skewY()`.

הערך translate מזיז את האובייקט ממקומו המקורי, ניתן להעביר לו ערך מספרי אחד או שתיים, כשנעביר לו ערך מספרי אחד, האובייקט יזוז רק בכיוון האופקי, על ציר ה-x, וכשניתן לו שתי ערכים, הוא יזוז גם בכיוון האנכי, הערך הראשון מציין את התזוזה על ציר ה-x והערך השני מציין את התזוזה על ציר ה-y.

לדוגמא:

`transform:translate(20px,10px);`

פקודה זו תזיז את האלמנט ממקומו המקורי 20 פיקסלים בציר ה-x ו-10 פיקסלים בציר ה-y.

ניתן לתת מספר ערכים לפעולת `transform` אחת.

לדוגמא:

`transform: translate(20px,10px) rotate(30deg) skew(20deg);`

בדוגמא זו קבענו מספר ערכים לאותה פעולת שינוי.

תרגיל 91

1. צרו מסמך html חדש
2. הוסיפו במסמך 3 אלמנטי div.
3. קבעו לכל אלמנט div צבע רקע כרצונכם.
4. קבעו לפסקה הראשונה סיבוב בזווית של 80 מעלות.
5. קבעו לפסקה השנייה הטיה בזווית של 45 מעלות.
6. קבעו לפסקה השלישית שתגדל פי 3 מגודלה המקורי.

התוכנה transition - פעולת השתנות

התכונה transition מאפשרת לשנות את האלמנט במצב מסוים, כגון בעת מעבר עכבר, תכונה זו היא מייצרת כעין אנימציות וקובעת את משך הזמן שבו מתבצע השינוי.

מבנה תכונה תכונת transition

תכונת transition בנויה בצורה הבאה:

Transition: <משך זמן> <מאפיין>

הערך הראשון הוא התכונה אותה רוצים לשנות, כגון תכונת אטימות, תכונת רוחב ועוד. דוגמא לשינוי רוחב אלמנט מרוחב של 100% לרוחב של 50% פיקסלים במשך 2 שניות נשתמש בקוד הבא:

```
#one{ background:red; transition: width 2s;}
```

```
#one:hover{ width:50%;}
```

בדוגמא זו בשורה הראשונה הגדרנו את השינוי בתכונה transition לתכונה width - רוחב, הגדרנו את משך הזמן ל 2 שניות, בשורה השנייה הגדרנו שבעת מעבר עכבר, האלמנט מקבל רוחב של 50%, בצורה זו יצרנו אנימציה בעת מעבר עכבר על האלמנט, אנימציה שמצרה את רוחב האלמנט לחצי מרוחבו המקורי.

שינוי מספר תכונות לאלמנט אחד

אם נרצה לשנות מספר תכונות לאלמנט אחד נציין כל אחת מהתכונות שאנחנו רוצים לשנות, לכל תכונה נציין את משך הזמן, בין תכונה לתכונה נשים את התו פסיק (,).

לדוגמא:

```
#two{ background:green; transition: background 2s, color 1s, transform:scale 1s opacity 1.5s;}
```

```
#two:hover{ background:yellow; color: red; transform:scale(2); opacity: 0.5; }
```


בדוגמא זו הגדרנו שינוי גם של צבע רקע, גם של צבע טקסט, גם של גודל האלמנט וגם של אטימות האלמנט.

לסיכום: על מנת לגרום לשינוי של אלמנט יש להגדיר שני דברים:

1. מגדירים פעולת השתנות באמצעות התכונה `transform` מגדירים את התכונות שמשתנות ומשך הזמן לכל שינוי תכונה.
2. מגדירים ערך חדש לתכונות שקבענו שמשתנות לאלמנט במצב של פוקוס או מעבר עכבר.

transition-delay

התכונה **transition-delay** מגדירה עיכוב, השהייה טרם תחילת האנימציה, תכונה זו מקבלת כערך ערך מספרי המוצמד לאות `s` (Seconds), ערך זה מציין את משך הזמן של עיכוב תחילת פעולתה של האנימציה במספר שניות, ערך ברירת מחדל של תכונה זו הוא 0 שניות, כך שאם לא נציין את התכונה, או שניתן לה את הערך 0, האנימציה לא תתעכב לפני פעולתה.

לדוגמא:

```
div {width: 100px; height: 100px; background: red; transition: width 3s; transition-delay: 1s;}
```

```
div:hover { width: 300px;}
```

בדוגמא זו מוגדרת השהייה של שנייה לפני תחילת פעולת השינוי של רוחב האלמנט.

transition-duration

התכונה **transition-duration** מגדירה את משך הזמן של האנימציה בשניות, כאשר נצמיד לאובייקט אנימציה נגדיר לו את התכונה **transition-duration**, תכונה זו מקבלת כערך ערך מספרי המוצמד לאות `s` (Seconds), ערך זה מציין את משך הזמן של האנימציה במספר שניות. ערך ברירת מחדל של תכונה זו הוא 0 שניות, מסיבה זו חובה לתת ערך מספרי הגדול מ-0 וכאשר לא נגדיר תכונה זו לאובייקט שהצמדנו לו אנימציה, האנימציה לא תופעל כלל.

transition-property

בתכונה זו נעביר את התכונה שאותה נרצה לשנות, לדוגמא:

```
div {  
  width: 200px;  
  height: 200px;  
  background: green;  
  transition-property: width;  
  transition-duration: 1s;  
}
```

```
div:hover { width: 300px;}
```

transition-timing-function

התכונה **transition-timing-function** מגדירה את קצב האנימציה, ניתן להגדיר קצב איטי בהתחלה או בסוף, קצב מהיר בהתחלה או בסוף.

הערכים שניתן להגדיר לתכונה זו:

ease - הגדרת ערך זה תגדיר לאנימציה להתחיל ולסיים באיטיות.

linear - הגדרה זו תגדיר לאנימציה לעבוד במהירות אחידה לכל אורכה.

ease-in - הגדרה זו תגדיר לאנימציה התחלה איטית.

ease-out - הגדרה זו תגדיר לאנימציה סיום איטי.

ease-in-out - הגדרה זו תגדיר לאנימציה איטיות באמצעה.

cubic-bezier(n,n,n,n) - ערך שהוא פונקציה המקבלת ארבעה ערכים מספריים המציינים את מהירות האנימציה בארבעה מצביה, תחילה, אמצע התחלתי, אמצע סופי וסיום.

לדוגמא:

transition-timing-function: cubic-bezier(0.1, 0.7, 1.0, 0.1);

כתיב מקוצר

הערה: פעמים רבות האלמנט שהגדרנו לו שינוי, מרצד בעת מעבר העכבר כשמבצע את פעולת השינוי, ריצוד זה נגרם מהסיבה שמעבר העכבר מתבצע על האלמנט, תוך כדי פעולת השינוי עדיין מתבצע שוב ושוב מכיון שהעכבר עדיין עובר על האזור שלו, כך שהוא מנסה לבצע שוב את הפעולה כשהוא עסוק בפעולתו הקודמת, לעומת זאת אם האלמנט היה עטוף באלמנט אב שפעולת מעבר העכבר מוגדרת עליו, ופעולות השינויים מוגדרים על האובייקט, מעבר העכבר היה מתבצע על אובייקט האב והאובייקט הפנימי מבצע את פעולתו בחלקלקות יותר.

לדוגמא:

נסו להגדיר לפסקה ברוחב של 100% סיבוב של 90 מעלות, כאשר עוברים על הפסקה

```
#four{ background:orange; transition-property: transform 2s;}
```

```
#four: hover { transform:rotate(90deg); }
```

כאשר נעבור על הפסקה, הפסקה תקבל את השינוי הזוויתי ותסתובב 90 מעלות, אך מכיוון שנעמוד על הפסקה, היא תרצד ותבצע את הפעולה שוב ושוב תוך ריצוד.

לעומת זאת אם נעטוף את הפסקה באלמנט `div`, ונגדיר את פעולה ההשתנות במעבר עכבר על האובייקט `div`.

```
#four{ background:orange; transition-property: transform 2s;}
```

```
div: hover #four{ transform:rotate(90deg); }
```

נשים לב להבדל שהשינוי יחול בצורה יותר עדינה וללא הריצוד שנראה קודם לכן.

תרגיל 92

1. בנו מסמך html המכיל 5 פסקאות.
2. כתבו בכל פסקה את אחד מהערכים הבאים: 50%, 33%, 25%, 70%, 80%.

לדוגמא:

<p>33%</p>

3. לכל פסקה קבעו צבע רקע שונה.
4. קבעו לכל הפסקאות רוחב של 100%.
5. קבעו שבעת מעבר עכבר על הפסקה, הפסקה משנה את גודלה בהתאם לטקסט המתואר בה.
6. קצב השינוי יהיה 2 שניות.

תרגיל 93

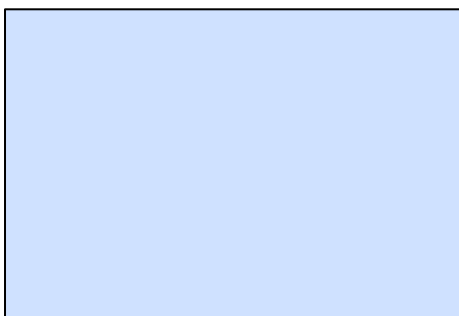
1. בנו מסמך html המכיל אובייקט div.
2. מרכזו את תוכן המסמך.
3. קבעו לו רוחב וגובה של 200 פיקסלים.
4. קבעו לו צבע רקע ירוק.
5. קבעו שבעת מעבר עכבר האובייקט div משנה את רוחבו וגובהו ל300 פיקסלים, ומסתובב בסיבוב של 180 מעלות.
6. הגדירו השהייה של שנייה לפני תחילת פעולת השינוי.

התכונה overflow

התכונה `overflow` מאפשרת להגביל חריגה של אובייקט פנימי מגבולות האובייקט החיצוני.

לדוגמא קיימת תיבת טקסט הממוקמת בתוך אלמנט `div`, הגדרנו לאלמנט `div` תכונת `overflow` מסוג `visible` ותכונת `width`, כאשר תיבת הטקסט תתארך מעבר לגובה המוגדר לתיבה `div` יראה שהטקסט חורג מגבולות המסגרת, המסגרת תעבור בתוך הפסקה, או אם מקמנו תמונה בתוך אובייקט עוטף יתכן שהתמונה תגלוש מגבולות אובייקט האב, ותראה בלתי קשורה.

דוגמא לתצוגת חריגת טקסט מגבולות התיבה בתצוגת הדפדפן:



על מנת לגבור על בעיה זו נשתמש בתכונה `overflow`, תכונה זו מגדירה לאובייקט האב, האובייקט החיצוני העוטף, שכל האובייקטים שממוקמים בו לא יגלוש מגבולותיו.

לתכונה זו ניתן להגדיר מספר ערכים:

visible - ערך ברירת המחדל המציין שהאובייקטים הפנימיים יוכלו לגלוש מגבולות האובייקט החיצוני.

hidden - ערך זה מגדיר שהתוכן שגולש מגבולות האובייקט החיצוני יוסתר ולא יוצג.

scroll - ערך זה מגדיר שהתוכן שגולש מגבולות האובייקט החיצוני יוסתר ולא יוצג, אך יתווספו סרגלי גלילה: אופקי ואנכי המאפשרים לצפות בחלקים המוסתרים, סרגלי הגלילה יתווספו בכל מקרה, גם כאשר הם מיותרים (אין תוכן מוסתר, או שיש תוכן מוסתר רק בכיוון אחד, אופקי או אנכי).

auto - ערך זה מגדיר שהתוכן שגולש מגבולות האובייקט החיצוני יוסתר ולא יוצג, אך יתווספו סרגלי גלילה רק בהתאם לצורך, אם אין צורך בסרגלי הגלילה הם לא יתווספו, אם

יש צורך רק בסרגל אופקי יתווסף רק סרגל אופקי, אם יש צורך רק בסרגל אנכי יתווסף רק סרגל אנכי.

תרגיל 94

1. צרו מסמך המכיל div ובתוכו טקסט ארוך.
2. קבעו לאלמנט div גובה של 500 פיקסלים.
3. קבעו שהטקסט הגולש מגבולות ה-div יוסתר.
4. קבעו פסי גלילה לפי הצורך.
5. צפו במסמך בדפדפן ובדקו שהכל עובד כראוי

תרגיל 95

1. הוסיפו תמונה בתוך אלמנט div.
2. הגדירו שהתמונה תהיה מעט גדולה מהאלמנט div.
3. קבעו שהתמונה תהיה במיקום יחסי לאלמנט div, כך שהתמונה תמוקם מתחת לאלמנט div.
4. קבעו שהתמונה הגולשת מהאלמנט תוסתר
5. בעת מעבר עכבר על הdiv, קבעו שהתכונה overflow תשתנה, כך שהתמונה תוצג.

אנימציות ללא תלות בפעולה

עד ל-css3 שילוב אנימציה ב-html היה דבר די מורכב, שילוב אנימציה התבצע באמצעות פיתוח ב-javascript או ב-flash. החל מגרסת css3 נוספה תמיכה באנימציות, מה שמקל מאד להוסיף אנימציה באמצעות CSS בלבד, האנימציה מאפשרת לשנות סגנון אלמנט, פעולת האנימציה מתבצעת שוב ושוב ללא עצירה, פעולת האנימציה אינה תלויה בפעולה כלשהי כמו מעבר עכבר לדוגמא.

הגדרת אנימציה נעשית באופן הבא:

```
@keyframes nameanimate{
```

```
From(נקודת מוצא);
```

```
To(יעד);
```

```
}
```

תחילה מופיעה המילה `@keyframes` ולאחריה מופיע שם האנימציה, בין הסוגריים המסולסלות, יבואו ההוראות המגדירות את פעולת האנימציה.

המילה from - מגדירה נקודת ההתחלה מאיזו נקודה האנימציה תתחיל, מה יהיה מצבה של התכונה שמקבלת שינוי בתחילת השינוי.

המילה to - מגדירה נקודת הסיום שאליה שואפת האנימציה להגיע, מה יהיה מצבה של התכונה שמקבלת שינוי בסיום האנימציה.

האנימציה מבצעת את כל שלבי הבינים החל מנקודת המוצא וכלה בנקודת היעד.

קישור האנימציה לאובייקט ספציפי

התכונה `animation-name`:

התכונה `animation-name` מגדירה את הצמדת האנימציה לאובייקט ספציפי, כאשר נרצה להצמיד לאובייקט אנימציה מסוימת נגדיר לו את התכונה `animation-name`, כשתכונה זו מקבלת כערך את שם האנימציה, שם האנימציה מופיע בשורת הגדרת האנימציה לאחר המילה `@keyframes`.

התכונה **animation-duration**:

התכונה **animation-duration** מגדירה את משך הזמן של האנימציה בשניות, כאשר נצמיד לאובייקט אנימציה נגדיר לו את התכונה **animation-duration**, תכונה זו מקבלת כערך ערך מספרי המוצמד לאות s (Seconds), ערך זה מציין את משך הזמן של האנימציה במספר שניות. ערך ברירת מחדל של תכונה זו הוא 0 שניות, מסיבה זו חובה לתת ערך מספרי הגדול מ-0 וכאשר לא נגדיר תכונה זו לאובייקט שהצמדנו לו אנימציה, האנימציה לא תופעל כלל.

התכונה **animation-delay**:

התכונה **animation-delay** מגדירה עיכוב, השהייה טרם תחילת האנימציה, תכונה זו מקבלת כערך ערך מספרי המוצמד לאות s (Seconds), ערך זה מציין את משך הזמן של עיכוב תחילת פעולתה של האנימציה במספר שניות, ערך ברירת מחדל של תכונה זו הוא 0 שניות, כך שאם לא נציין את התכונה, או שניתן לה את הערך 0, האנימציה לא תתעכב לפני פעולתה.

ניתן להגדיר גם ערך שלילי, כאשר נגדיר ערך שלילי האנימציה תתחיל כאילו היא כבר פעילה במשך מספר השניות שנגדיר.

לדוגמא:

`animation-delay: -2s;`

אם האנימציה מוגדרת למשך זמן של 5 שניות, היא תתחיל מהנקודה בה היא הייתה עומדת בשנייה מספר 3.

התכונה **animation-iteration-count**:

התכונה **animation-iteration-count** מגדירה את מספר האיטרציות של האנימציה, מגדירה את מספר הפעמים שהאנימציה תרוץ, ערך ברירת המחדל לתכונה זו הוא 1, כך שאם לא נגדיר תכונה זו האנימציה תפעל פעם אחת בודדת.

הערכים שניתן להגדיר לתכונה זו:

ערך מספרי - ערך הקובע את מספר הפעמים שהאנימציה תתבצע.

הערך infinite - אין סופי, ערך זה קובע שהאנימציה תרוץ שוב ושוב ללא מעצור.

לדוגמא:

```
div {  
  width: 100px;  
  height: 100px;  
  position: relative;  
  background-color: green;  
  animation-name: example;  
  animation-duration: 3s;  
  animation-iteration-count: 5;  
}
```

בדוגמא זו התכונה `animation-iteration-count` מגדירה לאנימציה שתפעל 5 פעמים.

התכונה `animation-direction`:

התכונה `animation-direction` מגדירה את כיוון האנימציה, ערך ברירת מחדל של תכונה זו הוא `normal`, האנימציה תעבוד בכיוון הרגיל בה היא מוגדרת תתחיל מנקודת ההתחלה (`from`) עד ההוראה נקודת הסיום (`to`), ניתן להגדיר גם שהאנימציה תפעל בסדר הפוך ותתחיל מנקודת הסיום לכיוון נקודת ההתחלה.

הערכים שניתן להגדיר לתכונה זו:

reverse - הגדרת ערך זה תגדיר לאנימציה לעבוד בכיוון ההפוך מהכיוון המוגדר לה, האנימציה תתחיל את פעולתה מהמוגדר בנקודת הסיום לכיוון נקודת המוצא.

alternate - הגדרת ערך זה תגדיר לאנימציה לעבוד בכיוון הרגיל ולאחר מכן בכיוון ההפוך, כך שוב ושוב עד לסיום פעולת האנימציה.

alternate-reverse - הגדרת ערך זה תגדיר לאנימציה לעבוד מהכיוון ההפוך לכיוון הרגיל ומהכיוון הרגיל לכיוון ההפוך וחוזר חלילה עד לסיום פעולת האנימציה.

דוגמא:

```
div {  
  width: 100px;  
  height: 100px;
```

```
position: relative;
background-color: green;
animation-name: example;
animation-duration: 3s;
animation-direction: reverse;
}
```

בדוגמא זו המאפיין **animation-direction** מגדיר לאנימציה שתפעל מהכיוון הסופי לכיוון ההתחלתי.

התכונה **animation-timing-function**:

התכונה **animation-timing-function** מגדירה את קצב האנימציה, ניתן להגדיר קצב איטי בהתחלה או בסוף, קצב מהיר בהתחלה או בסוף.

הערכים שניתן להגדיר לתכונה זו:

ease - הגדרת ערך זה תגדיר לאנימציה להתחיל ולסיים באיטיות.

linear - הגדרה זו תגדיר לאנימציה לעבוד במהירות אחידה לכל אורכה.

ease-in - הגדרה זו תגדיר לאנימציה התחלה איטית.

ease-out - הגדרה זו תגדיר לאנימציה סיום איטי.

ease-in-out - הגדרה זו תגדיר לאנימציה איטיות באמצעה.

cubic-bezier(n,n,n,n) - ערך שהוא פונקציה המקבלת ארבעה ערכים מספריים המציינים את מהירות האנימציה בארבעה מצביה, תחילה, אמצע התחלתי, אמצע סופי וסיום.

לדוגמא:

```
transition-timing-function: cubic-bezier(0.1, 0.7, 1.0, 0.1);
```

דוגמא:

קוד האנימציה:

```
@keyframes example {
  from {background-color: red;}
```

```
to {background-color: yellow;}  
}
```

האלמנט שעליו מוכלת האנימציה:

```
div {  
  width: 200px;  
  height: 200px;  
  background-color: red;  
  animation-name: example;  
  animation-duration: 3s;  
}
```

אנימציה זו קובעת שינוי צבע לאלמנט מצבע אדום לצהוב.

באלמנט המקבל את האנימציה מוגדר שהאנימציה תמשך במשך שלוש שניות.

שימו לב: ניתן להשתמש באותה אנימציה למספר אובייקטים שונים.

מספר שלבים באנימציה

ניתן לתת מספר שלבים באנימציה, על מנת ליצור אנימציה יותר מורכבת ומעניינת.

את שלבי האנימציה נגדיר באחוזים, לדוגמא אם נרצה לתת לאנימציה ארבע שלבים, נגדיר ארבע נקודות עצירה באחוזים, נקודות העצירה יכולות להיות בחלוקה זהה של 25% לכל חלק, כך שהחלק הראשון יהיה 0% לאחריו 25% לאחריו 50% ולאחריו 100%, ניתן לחלק את נקודות העצירה באופנים רבים לאו דווקא שווים.

לדוגמא:

קוד האנימציה

```
@keyframes example {  
  0% {background-color: red;}  
  35% {background-color: orange;}
```

```
50% {background-color: green;}
100% {background-color: blue;}
}
```

קוד החלת האנימציה על האלמנט

```
div {
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

כתיב מקוצר

ניתן לכתוב את כל תכונות האנימציה בקצרה כששם התכונה הוא animation והערכים שהיא תקבל הם: שם האנימציה, משך הזמן, קצב האנימציה, פרק זמן השהייה (אם קיימת השהייה), מספר פעמים שהאנימציה תופיע וכיוון האנימציה.

לדוגמא:

```
div {
  animation-name: example;
  animation-duration: 3s;
  animation-timing-function: linear;
  animation-delay: 1s;
  animation-iteration-count: infinite;
  animation-direction: alternate;
}
```

את הגדרות האנימציה הללו ניתן להגדיר בשורה אחת:

```
div {  
  animation: example 3s linear 1s infinite alternate;  
}
```

תרגיל 96

1. בנו באנר המחליף תמונות רקע של צמחים. (השתמשו בתכונה background-image)
2. הבאנר יורכב מאנימציה המחליפה לפחות 5 תמונות (5 נקודות עצירה).
3. הגדירו את משך הזמן של האנימציה ל35 שניות.
4. קבעו שאנימציה תתחיל ותסיים בקצב איטי.
5. קבעו שהאנימציה תעבוד שוב ושוב ללא עצירה.

java script הקדמה, משתנים, השמת ערכים למשנים וחישובים בסיסיים

Js היא שפת תכנות לאתרים, שפות התכנות המטפלות באתרים מתחלקות לשני סוגים שפות backend ו-frontend, כששפות ה-backend הן שפות הצד שרת, מעבדות ומטפלות בנתונים המועברים לשרת, ומתקשרות עם השרת, מעדכנות נתונים במס נתונים בשרת, שולפות נתונים ממסד הנתונים בשרת ושולחות אותם לתצוגה לגולש. ושפות ה-frontend הן השפות שרצות בצד הלקוח על ידי הדפדפן. שפת הג'אווה סקריפט היא שפת לקוח, היא מרחיבה את יכולות שפת html ומאפשרת ליצור יישומי אינטרנט מתוחכמים ומשוכללים.

שפה זו פותחה בשנת 1995 שפת גאווה סקריפט היא גם שפה מונחת אירועים, כשכל מהלך ריצת התוכנית מבוסס על אירועים שמתרחשים, לכל אובייקט קיים אוסף של אירועים אפשריים, כמו לחיצת עכבר, מעבר עכבר, פוקוס ועוד.

שפת javascript היא שפה רגישה לאותיות גדולות וקטנות, כך שהמילה new והמילה NEW מבחינתה הן שתי מילים שונות לחלוטין.

מה ניתן לעשות בjs?

ניתן באמצעות js לבדוק חוקיות של קלט שהמשתמש הזין, ליצור פופאפים למשתמש, להציג או להסתיר דברים, לדחוף קלאסים לאלמנטים על מנת לקבל עיצוב כזה או אחר ועוד ...

שלושה מקומות לכתיבת js באתר:

ניתן לשלב את קוד ה-JavaScript בשלושה אופנים:

1. **בגוף הדף** - בתוך התגית script, ניתן לשלב קטעי JavaScript, את תגית ה-script ניתן להניח ב-head או ב-body, בתגית הפותחת של תגית ה-script נגדיר שזהו סקריפט של javascript באמצעות המאפיין type כך: "type=text/javascript".

כאשר נגדיר את קטעי ה-javascript בתוך תגית ה-script ב-head לא נוכל לגשת לאלמנטים שעדיין לא קיימים, הדפדפן קורא את הקוד מתחילתו עד סופו, כך שאם הגדרנו את קוד ה-javascript בתוך תגית script בתגית head, הדפדפן קורא קוד זה לפני שקורא את תוכן הדף, את תוכן התגיות שבהמשכו של הדף, אם נתייחס לאלמנטים בדף, הדפדפן לא יציג את תוצאת הקוד שמתייחס לאלמנטים שעדיין לא קיימים מבחינתו.

דוגמא לשילוב js בגוף מסמך ה-html:

```
<!DOCTYPE html>
<html>
<head>
  <title>js print</title>
  <style type="text/css">
  </style>
  <script type="text/javascript">
    alert("שלום גאוה סקריפט");
  </script>
</head>
<body dir="rtl">
<p id="demo"></p>
</body>
</html>
```

בקוד זה הגדרנו פופאפ שיקפוץ עם פתיחת הדף, בפופאפ כתבנו שלום גאוה סקריפט. בדפדפן זה יראה כך:



2. בגוף האלמנט - בתוך תגית html, ניתן לשלב פקודות וקודי java script, שיטה זו לא מומלצת כלל, היא לא מאפשרת מעקב נח אחרי פקודות ושגיאות, ולכל עדכון של קטעי קוד חופפים יש צורך לעבור בין כל האלמנטים, במקום לעדכן במקום אחד.

דוגמא:

```
<!DOCTYPE html>

<html>

<head>

    <title>js print</title>

</head>

<body dir="rtl">

    <button                                type="button"
onclick="document.getElementById('demo').innerHTML = Date()">
לחץ וקבל את השעה והתאריך הנוכחיים
</button>

<p id="demo"></p>

</body>

</html>
```

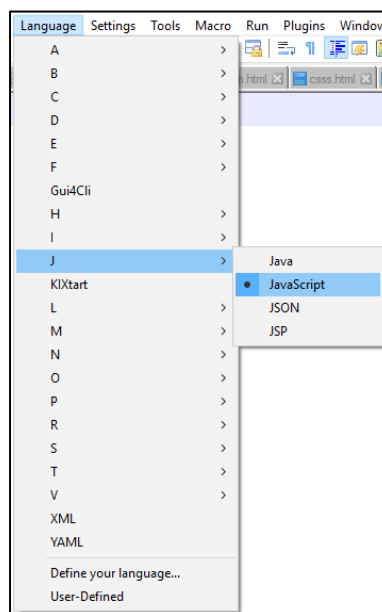

בקוד זה הגדרנו בגוף האלמנט `button` פקודת `JavaScript` המתבצעת בעת לחיצה, הפקודה הוגדרה כערך למאפיין `onclick` והדפיסה בעמוד בתצוגת הדפדפן את השעה והתאריך הנוכחיים.

3. **קובץ חיצוני** - כשם שלמדנו בשילוב ה-`CSS` במסמך, גם קבצי גאווה סקריפט, הדרך המומלצת לעבודה איתם היא, הפרדת הפקודות לקובץ נפרד, וקישור דפי ה-`html` המשתמשים בקוד, על ידי קישור של הקובץ בצורה זו, אנו מורים לדפדפן להריץ את קובץ ה-`javascript` בכל טעינה מחדש.

על מנת לקשר מסמך `html` לקובץ גאווה סקריפט, יש להוסיף קישור לקובץ בתגית `script` את תגית הסקריפט אפשר למקם בתגית ה-`head` או בתגית ה-`body`.

יצירת מסמך `js`

את מסמכי ה-`javascript` כמו מסמכי ה-`css` ניצור בעורך הטקסט, נגדיר בתפריט `language` שהקובץ שלנו הוא קובץ `javascript`



שמירת הקובץ תבצע כשמירת כל קובץ: `file>save as...`

בסיום שמירת הקובץ נשים לב שנוצר לנו קובץ עם הסיומת `.js`.

הפקודה alert מקפיצה חלון פופאפ לדוגמא:

```
alert("שלום גאווה סקריפט");
```

יקפיץ חלון פופאפ עם הכיתוב שלום גאווה סקריפט.

1. צרו מסמך html חדש.
2. שלבו את הפקודה alert() בתגית script, כתוב בתוכה "שיעור ראשון ב-javascript".
3. צרו מסמך html נוסף שמור אותו בשם tar2.
4. צרו מסמך js מקום בתוכו את הפקודה alert(); כתוב בתוכה "שיעור ראשון ב-javascript".
5. קשרו את המסמך tar2 לקובץ ה-js שיצרת.
6. הפעילו את שני הקבצים ובדקו שהם עובדים כראוי.

מספר דרכים להצגת פלט js

פלט הכוונה לקבל משוב של תוצאה, לקבל תגובתיות מהתוכנית הכתובה.

1. הפקודה alert - פקודה זו פותחת חלון קטן קופץ בראש חלון הדפדפן ובו הודעת טקסט.

לדוגמא:

```
Alert("שלום עולם");
```

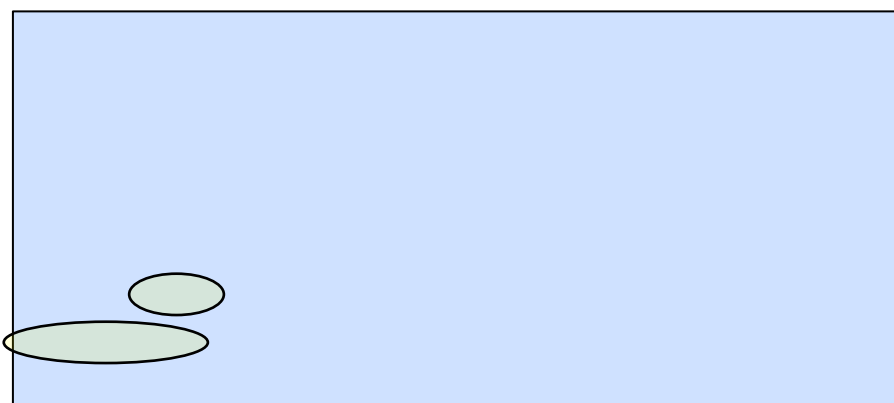


2. **פלט לחלון ה-console** - חלון ה-console יוצג כאשר נכנס למצב "בדוק מרכיב" (לחיצה ימנית על העכבר בדף פתוח בדפדפן < בדוק מרכיב או לחיצה על מקש הקיצור f12).

לדוגמא:

```
<!DOCTYPE html>
<html>
<head>
  <title>js print</title>
  <script type="text/javascript">
    console.log("שלום גאווה סקריפט");
  </script>
</head>
<body dir="rtl">
<p id="demo"></p>
</body>
</html>
```

את פלט הקוד נוכל לראות בחלון ה-console, תצוגתו תראה כך:



את חלון ה-console ניתן לראות כאשר נכנסים למצב של בדיק מרכיב תחת הלשונית .console

לחלונית ה-console ניתן לשלוח מספר מחרוזות, מספר טקסטים כשפסיק מפריד בניהם.

3. כתיבה נתונים לאמנטים ב-html באמצעות innerHTML

ניתן לגשת לאמנטים ב-html ולהכניס לתוכם תוכן באמצעות פונקציית javascript, המקבלת את שם ה-id של האלמנט ומכניסה לתוכו תוכן, התוכן שיוכנס לאלמנט יוגדר לאחר הסימן שווה בתוך גרשיים. הפונקציה נראית כך:

```
document.getElementById('demo').innerHTML = "כך נכניס תוכן לאמנטים = " בדף ההטמל
```

לדוגמא:

```
<!DOCTYPE html>
<html>
<head>
  <title>js print</title>
</head>
<body dir="rtl">
  <p id="demo"></p>
  <script type="text/javascript">
    כך נכניס תוכן = "document.getElementById('demo').innerHTML = " בדף ההטמל
    </script>
  </body>
</html>
```

קוד זה ידפיס לנו בפסקה עם המזהה demo את הטקסט "כך נכניס תוכן לאמנטים בדף".

משתנים

משתנים הם כעין תאי אחסון בזיכרון, בהם ניתן ל"אחסן" ערכים.

על מנת ליצור תא אחסון לשמירת ערכים יש צורך להגדיר משתנה, המשתנה הוא המצביע לתא האחסון.

על מנת ליצור משתנה נשתמש בפקודה `var` או `let` (קיצור המילה `variable` - משתנה).

דוגמא ליצירת משתנה:

```
var x;
```

ניתן גם ליצור משתנה באמצעות הפקודה `let`, בעזרת הפקודה `let`.

דוגמא ליצירת משתנה:

```
let x;
```

שמות המשתנים

שמות המשתנים מורכבים מאותיות באנגלית, ניתן לצרף למשתנה מספרים, אך משתנה שם משתנה לא יתחיל במספר, משתנה הוא מילה אחת, אם נרצה להגדיר משתנה בעל מספר מילים, נשתמש בקו תחתי בין המילים השונות, או שנכתוב את צירוף המילים ללא רווח ביניהם, כשמילה חדשה מתחילה באות גדולה, שם משתנה לא יכול להכיל רווחים.

בשורה שכתבנו הגדרנו בזיכרון תא בשם `x`, מרגע הגדרתו התא נוצר ונקבע עבורו מקום בזיכרון, התא `x` כרגע ריק מכל תוכן, נוכל להכניס בתוכו מספרים או מחרוזות טקסטים, נוכל במהלך תכנית ה-`js` לשנות את ערכו, על ידי חיבור של ערכים נוספים אליו.

חשוב לתת לשמות שם בעל משמעות, על מנת שנוכל לעקוב אחר התוכנית בנוחות, ולהבין את משמעות המשתנים גם לאחר זמן.

כמה כללים

כל פקודה ב-`javascript` מסתיימת בסימון פסיק נקודה (`;`).

שילוב הערות בקוד

ניתן לשלב הערות בקוד ה-js על מנת לתעד את התהליכים שבצענו בקוד, או לציין הערות שונות בגוף הקוד.

שילוב הערה המיועדת לעורך הקוד, ואינה מוצגת למשתמשים תתבצע באמצעות הסימון: לוכסן לוכסן (//) להערה באורך של עד שורה אחת.

הערות המתפרסות על פני מספר שורות תתחלנה בסימן לוכסן כוכבית (*/) ובסיום ההערה לאחר מספר שורות יבוא הסימון (*/) לסגירת ההערה.

לדוגמא:

```
// זו הערה של שורה אחת/
```

```
/* זו הערה */
```

```
המתפרסת על פני כמה
```

```
שורות ומסתיימת
```

```
*/בכוכית לוכסן
```

לעיתים רבות נשתמש בהערות בקוד גם למטרת נטרול חלקים מתוכנית ה-js, להתמקדות במקור תקלה או ל"הקפאת" חלקי קוד.

השמת ערכים במשתנה

לאחר שהגדרנו תא במשתנה נוכל לבצע פעולת השמה, לשים בו ערך, על מנת להכניס בו ערך נוסף את אופרטור ההשמה, הסימן שווה (=) בין שם המשתנה לערך אותו אנו מכניסים.

לדוגמא:

```
X=5;
```

בצורה כזו הכנסנו לתא x את הערך המספרי 5.

כאשר נכניס את הערך כך, מוקף בגרשיים:

```
X="5";
```

הכנסנו לתוך התא x את הערך "5", המשתנה x מחזיק בתוכו מחרוזת, כשערך המחרוזת הוא תו בודד "5".

משתנים מאפשרים לנו להשתמש שימוש חוזר באותם ערכים בלי לכתוב בכל פעם מחדש את הערך, כמו למשל המשתנה X יכול להכיל מחרוזת תווים ארוכה, כשבכל פעם שנרצה לכתוב את המחרוזת נכתוב בקוד שלנו רק את הערך x , ובכך נקבל את כל תכולתו המאוחסנת בו.

נוכל להכניס במשתנה גם תרגילים מתמטיים, כשהערך שישמר במשתנה הוא תוצאת החישוב.

לדוגמא:

```
X= 24+5;
```

הערך שנשמר בתא x הוא 29.

ניתן לסכום את ערכו של משתנה עם מספר נוסף לדוגמא תוצאת הפלט של הקוד הבא תציג את המספר 7:

```
let x=5;
```

```
x=x+2;
```

```
console.log(x);
```

ניתן לכתוב את הפעולה $x=x+2$; בקיצור ללא שימוש חוזר בערך x בכתיבת הפעולה כך:

```
X+=2;
```

האופרטור $+=$ מגדיר שאנחנו מחברים את ערך המשתנה x פלוס הערך שבא אחרי הסימן שווה.

סוגי ערכים במשתנים

הערך שנכניס למשתנים יכול להיות מסוגים שונים:

Number - מספר - מספר כלשהו חיובי או שלילי.

String - מחרוזת - טקסט המורכב מתו בודד או מספר תווים, הטקסט שנכניס למשתנה יוקף בגרשיים.

Boolean - ערך בוליאני - ערך המקבל את הערכים true או false (אמת או שקר).

אופרטורים מתמטיים:

+ חיבור

- חיסור

* כפל

/ חילוק

% שארית - כאשר נשתמש בסימן זה בחישוב נקבל את שארית החלוקה של החישוב לדוגמא:

$$51\%10=1$$

תוצאת תרגיל זה היא 1. שארית החלוקה של 51 ב-10.

+= הוספת ערך לעצמו

-= החסרת ערך מעצמו

בכל תרגיל שמרו את קוד ה-js בקובץ נפרד, קשרו מסמך html לקובץ ה-js.

תרגיל 98

1. כתבו תכנית המסכמת את סכום המספרים 48 ו-82,
2. הדפיסו את התוצאה לחלון ה-console.

תרגיל 99

1. כתבו תכנית המקבלת את הערך 68 למשתנה number.
2. חברו למשתנה number את הערך 258, עשו זאת באמצעות חיבור מקוצר (+=).
3. הדפיסו את התוצאה לחלונית alert. (ההדפסה תתבצע כך: `alert(number);`)

תרגיל 100

1. כתבו תכנית js, הגדירו בה משתנה חדש בשם number.
2. במשתנה number חשבו את ממוצע המספרים 100, 180, 190.
3. הדפיסו את התוצאה לחלון.

(הדפסה לחלון: `document.getElementById('demo').innerHTML = number`)

ניתן לקבל ערך למשתנה מהגולש באמצעות הפקודה `window.prompt`

הפקודה תראה כך:

```
num = window.prompt('הקש מספר');
```

בדוגמא זו הערך שהגולש יקיש ישמר במשתנה num.

תרגיל 101

1. קבל מהגולש שנת לידה.
2. חשב את שנת הלידה של הגולש (שנה נוכחית פחות שנת לידה).
3. הדפס לחלונית alert.

תרגיל 102

1. בכיתת תלמידים חולקה חבילה ובה 10 פרסים ל-4 ילדים

2. כתוב תכנית המחשבת את הפרסים הנותרים (השתמש ב%).

3. את התוצאה הדפס לחלון ה-console.

שרשור מחרוזות

ניתן לשרשר, לחבר מספר מחרוזות למחרוזת טקסט אחת ארוכה, שרשור מחרוזות נעשה באמצעות הסימן + בין מחרוזת אחת לשנייה, על מנת לקבל ריווחים בין השמות יש צורך לשרשר בנוסף את הריווח בין מילה למילה, ריווח יצוין בין גרשיים. כאשר הערכים במשתנים שנבצע עליהם את פעולת החיבור הם ערכים מספריים התוצאה שנקבל היא תוצאה מספרית.

תרגיל 103

1. כתבו תכנית js, הגדירו בתכנית 2 משתנים: firstName ו-lastName.
2. הערך שתכניסו במשתנה firstName הוא שמכם הפרטי.
3. הערך שתכניסו במשתנה lastName הוא שם משפחתכם.
4. הדפיסו הודעה בחלון קופץ, תוכן ההודעה "שלום שמכם הפרטי שם משפחתכם".
5. שימו לב לריווחים בין המחרוזות.

סדר פעולות חשבון

גם בתכנות יש חשיבות לסדר פעולות חשבון, ביטוי מתמטי יחושב משמאל לימין, כמובן, וקיימת קדימות לפעולות כפל וחילוק על פני חיבור וחסור, כמו כן ניתן לשלב בביטויים סוגריים בכדי לשנות את סדרן של הפעולות.

משפטי תנאי, ומשפטי תנאי מקוצרים

משפטי תנאי

קוד שכתבנו מבוצע על פי סדרו הכרונולוגי, על פי הסדר שהקוד כתוב, כך יתבצעו הדברים, לעיתים נרצה להתנות התנהגות מסוימת בתנאי אחד והתנהגות שונה בתנאי אחר, במקרים כאלו נשתמש בפעולת תנאי, כאשר התנאי יתקיים, הוא יחזיר את הערך true ותתבצענה הפעולות שקבענו בתנאי, כאשר התנאי לא יתקיים, הוא יחזיר את הערך false והקוד ימשיך לרוץ בלא התייחסות לפקודות בבלוק התנאי, ישנה אפשרות להניח בלוק קוד שיתבצע רק כאשר התנאי לא מתקיים.

אופרטורים המשמשים במשפטי תנאי:

== שווה שווה - השוואה בין שני ערכים, הסימן = פעם בודדת משמש להשמה, מכניס את הערך למשתנה עליו צוין, הסימן == משמש להשוואה, לדוגמא האם ערך משתנה מסוים ערכו זהה לערך משתנה אחר.

<= קטן או שווה

>= גדול או שווה

< קטן מ...

> גדול מ...

&& וגם רק כאשר שני התנאים מתקיימים הקוד בבלוק התנאי יתבצע.

|| או מספיק שתנאי אחד יתקיים בכדי שהקוד בבלוק התנאי יתבצע.

! not - לא .

!= - לא שווה, כלומר שונה.

מבנה משפט תנאי

```
If-אם (תנאי מתקיים) {  
  בלוק פעולות שיתבצע כשהתנאי מתקיים //  
}
```

דוגמה למשפט תנאי:

```
let x= hour;  
  
x= 16;  
  
if (hour < 18) {  
  print = "יום טוב";  
}
```

תרגיל 104

1. קבל מהגולש שנת לידה.
2. חשב את שנת הלידה של הגולש (שנה נוכחית פחות שנת לידה).
3. בדקו אם הגיל שקיבלתם גדול מ-18 הדפיסו למסך.

תרגיל 105

1. כתבו תכנית המקבל שעה בלבד.
2. בדקו אם השעה גדולה מ-08 הקפיצו חלוני alert "איחרת".

משפטי תנאי if-else

משפט תנאי if-else מאפשר לבצע קוד אחד במידה והתנאי מתקיים (ערך הביטוי true) וקוד שונה במידה והתנאי לא מתקיים (ערך הביטוי false).

הביטוי else משמעותו אחרת, לביטוי if else שני בלוקים, בלוק אחד לקוד במידה והתנאי מתקיים, במידה וה-if מחזיר ערך חיובי - true, ובלוק קוד נוסף במידה והתנאי לא מתקיים ה-if, התנאי מחזיר ערך שלילי - false, כל בלוק הוראות לביצוע יתחם בסימון סוגריים מסולסלות.

```
Let price = 1000;
```

```
If (price > 1500)
```

```
Console.log("יקר");
```

Else

```
Console.log("זול");
```

תרגיל 106

1. כתבו תכנית הקולטת מהמשתמש איזה יום היום.
2. אם המשתמש הקליד יום ראשון.
3. הדפיסו שבוע טוב.
4. אחרת הדפיסו היום לא יום ראשון.

תרגיל 107

מחיר מוצר חוקי הוא מחיר חיובי (גדול מאפס).

1. כתבו תכנית הבודקת אם מחיר מוצר הוא חוקי
2. אם המחיר חוקי הדפיסו לחלון ה-console "מחיר חוקי".
3. אחרת הדפיסו "המחיר אינו חוקי".

תרגיל 108

1. קלטו את שנת הלידה של המשתמש.
2. בדקו האם האדם אינו קטין (מעל גיל 18) וגם מתחת לגיל פרישה (גיל 65).
3. הדפיסו הודעה בהתאם.

תרגיל 109

1. בדקו האם היום יום זוגי (שני רביעי או שישי) (השתמשו באופרטור || - או)
2. הדפיסו הודעה בהתאם

תרגיל 110

1. קלטו מהמשתמש את שמו.
2. קלטו מהמשתמש את עיר מגוריו.
3. אם המשתמש הקליד "ירושלים".
4. הדפיסו את שם המשתמש + "אתה גר בעיר הבירה של ישראל".

5. אם המשתמש הקליד משהו אחר הדפיסו את שם המשתמש + "אינך גר בעיר הבירה של ישראל".

תרגיל 111

מספר זוגי הוא מספר אשר לא משאיר שארית בחלוקה ל-2.

1. כתבו תכני אשר קולטת מהמשתמש מספר שלם.
2. בדקו אם המספר שנקלט הוא מספר זוגי.
3. הדפיסו מספר זוגי.
4. אחרת הדפיסו המספר לא זוגי.

תרגיל 112

1. כתבו תכנית הקולטת מהמשתמש את גובה המשכורת שלו ואת גילו, אם גילו של המשתמש גדול מ-18 ומשכורתו החודשית עולה על 3200 שקלים, התכנית תנקה 6% למס בריאות.

2. התוכנית תפיס את משכורתו לאחר הניקוי.

תנאי בכתוב מקוצר

ניתן לכתוב גם תנאי בצורה מקוצרת כך:

```
If(a>5)? Alert("גדול"): alert("קטן");
```

אחרי התנאי מופיע הסימן שאלה, לאחריו יופיע מה יקרה אם התנאי מתקיים, ולאחר הסימן נקודתיים יתואר מה יקרה אם התנאי לא יתקיים.

Dom- Document Object Model

נגענו בבסיס של שפת javascript, על מנת לדעת את השפה על בוריה יש צורך בלימוד של עוד נושאים רבים כגון: לולאות, מערכים, פונקציות, ספריות מתמטיות ועוד.

טיפול במשתנים ומשפטי תנאי הם בסיס שבהחלט נוכל בעזרתם לגעת ולטפל בנתונים שונים באתר.

בפרק זה נדון על מבנה מסמך html ועל הדרך של תכנית javascript לשנות דברים בדף.

Dom ראשי תיבות של המילים Document Object Model - מודל אובייקט מסמך, זהו מודל תבנית סטנדרטית של מסמך html מודל העומד מאחורי תבנית דף html, המאפשר לשנות תוכן בדף באמצעות תכניות javascript, מודל זה מאפשר לגשת ולעדכן באופן דינמי את התכולה של הדף המבנה והסגנון של המסמך, תוך כדי טעינת הדף ולאחר שנטענו האובייקטים בדף.

ממשק ה-dom מייצג את המסמך כך שכל אלמנט במסמך מיוצג עם תכונות ואפשרויות פעולה, תכונות כמו צבע, גודל, אורך, רוחב, טקסט, אירועי לחיצה וכו', ממשק ה-dom הוא זה המאפשר לתוכנית לשלוט בתכונות הללו, ארגון התקינה w3c פרסם תקנים ל-dom הנקראים בשם w3dom.

דוגמא למבנה ה-dom למסמך document



שינוי תוכן ב-dom

ניתן לשנות תוכן של אלמנט ב-dom, כמה שיטות לגשת לאלמנט הספציפי לשינוי תוכנו:

1. גישה באמצעות ה-id - ניתן לגשת לאלמנט באמצעות ה-id שלו, תחילה ניגש למסמך הראשי הנקרא document בתוכו ניגש לאלמנט, נאתר אותו באמצעות ה-id הספציפי שלו, ונכניס לו ערך html-י.

נעשה זאת כך:

```
document.getElementById("demo").innerHTML = "טקסט חדש";
```

ניגשנו לאלמנט בעל המזהה demo, שינינו את הטקסט שלו והכנסנו לו את הכיתוב: "טקסט חדש".

נוכל כך גם להדפיס ערך של אלמנט בדף לפופאפ:

```
alert(document.getElementById("demo").innerHTML);
```

שורה זו תדפיס לפופאפ את תוכן האלמנט בעל המזהה demo.

2. גישה באמצעות tagName (שם תגית)

```
document.getElementsByTagName(name)
```

3. גישה באמצעות className (שם קלאס - מחלקה)

```
document.getElementsByClassName(name)
```

https://www.w3schools.com/js/js_html_dom_document.asp

עבודה עם אובייקטי ה-Dom היא עבוד מורכבת קריאה לכל אלמנט ומאפייניו נעשית בשורת גישה די ארוכה ומסובכת, על מנת ליעל את העבודה ולקצר אותה נבנתה ספרייה המבוססת על שפת JavaScript בשם jQuery, ספריית jquery זו ספרייה פופולארית מאד המאפשרת ליצור הרבה פונקציונליות במעט קוד. ניתן בעזרתה ליצור אפקטים ואנימציות שבאמצעות קוד JavaScript דורשים כתיבת קוד רבה וארוכה במספר שורות.

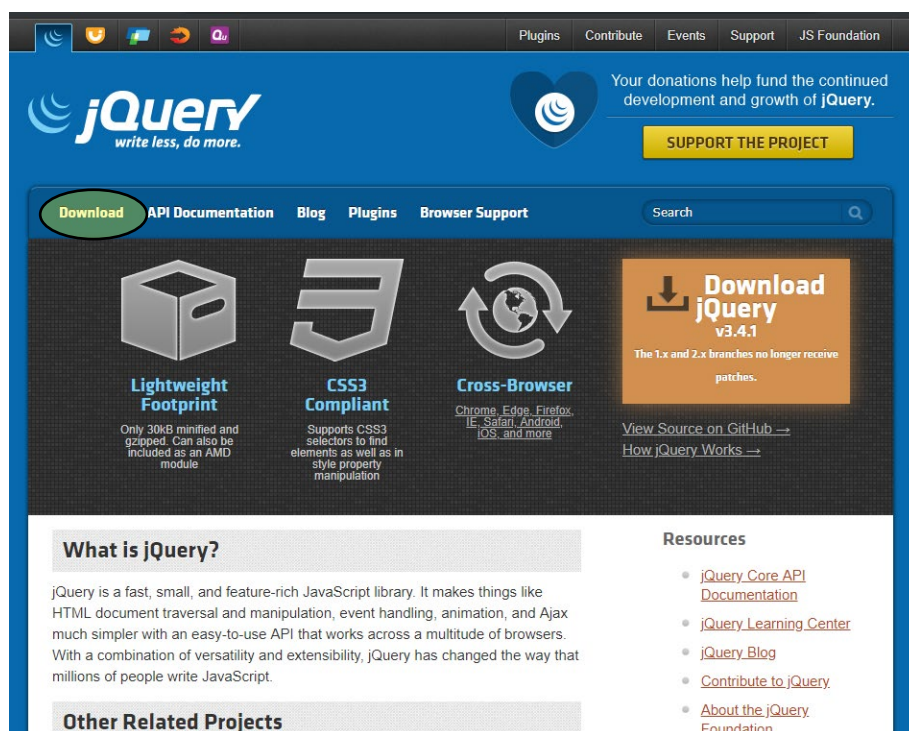
טעינת ספריית jquery

ספריית ה-Jquery צריכה להיטען באחד משתי המקומות, או ב-header או ב-body, בכל מקרה הספרייה חייבת להיטען לפני שנעשה בה שימוש בקוד ה-js, על מנת שהקוד שמתמש בה יהיה תקין.

שמירת הקובץ באתר וטעינתו

ניתן להוריד את הספרייה המלאה מהאתר של jQuery:

1. גלשו לאתר jQuery.com
2. לחצו על טאב ה-Download



3. בחרו בגרסת ה-jquery שברצונכם להוריד.

Download API Documentation Blog Plugins Browser Support Search

Downloading jQuery

Compressed and uncompressed copies of jQuery files are available. The uncompressed file is best used during development or debugging; the compressed file saves bandwidth and improves performance in production. You can also download a [sourcemap file](#) for use when debugging with a compressed file. The map file is *not* required for users to run jQuery, it just improves the developer's debugger experience. As of jQuery 1.11.0/2.1.0 the `///
sourceMappingURL` comment is **not included** in the compressed file.

To locally download these files, right-click the link and select "Save as..." from the menu.

jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relevant to your version. We also recommend using the [jQuery Migrate plugin](#).

[Download the compressed, production jQuery 3.4.1](#)

[Download the uncompressed, development jQuery 3.4.1](#)

[Download the map file for jQuery 3.4.1](#)

You can also use the slim build, which excludes the [ajax](#) and [effects](#) modules:

[Download the compressed, production jQuery 3.4.1 slim build](#)

[Download the uncompressed, development jQuery 3.4.1 slim build](#)

[Download the map file for the jQuery 3.4.1 slim build](#)

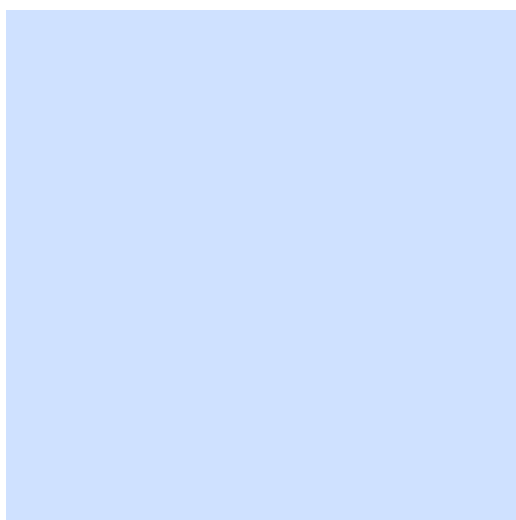
[jQuery 3.4.1 release notes](#)

Downloading jQuery using npm or Yarn

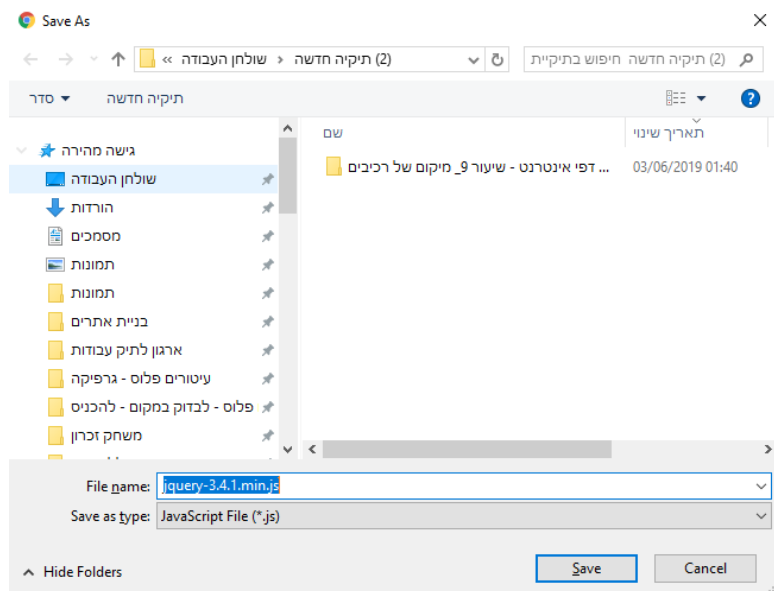
jQuery is registered as [a package](#) on [npm](#). You can install the latest version of jQuery with the npm CLI command:

4. אם ירד לכם קובץ js הניחו אותו באתרכם.

אם הופיע לכם תצוגה כזו



לחצו על המקשים ctrl+s לשמירת הקובץ, שמרו אותו בתיקיית אתרכם, שימו לב שהקובץ נשמר בסיומת .js.



לאחר ששמרתם את הקובץ באתרכם, טענו אותו לאתר, כשם שטוענים כל קובץ .js לדוגמא:

```
<script src="Scripts/jquery-3.4.1.min.js" type="text/javascript"></script>
```

שיטה זו עלולה להאט את האתר ולדרוש רוחב פס גדול על מנת לשרת את הגולשים בו.

טעינת הקובץ מהאתר החיצוני

ניתן גם לטעון את הקובץ הספרייה היושב באתר של-jQuery על ידי קישור אליו, הקישור יעשה באופן הבא:

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">  
</script>
```

השורה document ready

על מנת שהקוד יעבוד יש צורך לעטוף אותו בבלוק document ready שורה זו מעכבת את טעינת הקוד עד לסיום טעינת קוד מסמך ה-document, מסמך ה-html בכדי למנוע מצב שינסו לגשת לאלמנטים שטרם נוצרו, היא מעכבת את טעינת הפקודות עד שהמסמך מוכן.

דוגמא:

```
$(document).ready(function(){  
  // כאן יכתב הקוד שיבוצע לאחר שהדף מוכן  
});
```

ניתן לכתוב זאת גם כך:

```
$(document).ready(() => {  
  // כאן יכתב הקוד שיבוצע לאחר שהדף מוכן  
});
```

ביצוע פעולה ב-jquery - תחביר בסיסי

התחביר הבסיסי לביצוע פעולה כלשהי על האלמנטים ב-html הוא:

```
$(selector).פעולה();
```

הסימן דולר יתחיל כל פעולה, הסימן דולר הוא קיצור המגלם בתוכו את כל התחביר הבסיסי הנכתב במסמך javascript לגישה לאלמנט ב-dom, הוא יופיע לפני סוגריים, בסוגריים נציין את שם הסלקטור, כשם שכותבים סלקטור במסמך ה-css, שם הסלקטור יופיע בתוך גרשיים.

לדוגמא סלקטור למזהה ייחודי - id יכתב כך:

```
$("#idname").פעולה();
```

סלקטור ל-class מסוים יכתב כך:

```
$(".classname").פעולה();
```

סלקטור לתגית במסמך ה-html:

```
$(".classname").פעולה();
```

דוגמא לפעולה:

```
$("#p").hide();
```

פעולה זו תסתיר את כל הפסקאות במסמך ה-Dom, מסמך ה-html.

אירועי עכבר בjquery

הפעולות השונות שדף האינטרנט יכול להגיב עליהן מכונות אירועים, אירוע מייצג את הרגע שבו קורה משהו, לדוגמא: הזזת עכבר, בחירת לחצן, לחיצה על אלמנט, בפרק זה נתמקד באירועי עכבר.

click - אירוע זה מתרחש כאשר העכבר נמצא על אלמנט, לחץ עליו ושחררו.

Dbclick - אירוע זה מתרחש לאחר שהגולש לחץ לחיצה בעכבר.

Hover - אירוע זה מתרחש כאשר העכבר מרחף, עובר על אלמנט.

Mouseenter - אירוע זה מתרחש כאשר העכבר נכנס לתוך אלמנט מסוים, תיבת טקסט וכדומה.

Mouseleave - אירוע זה מתרחש כאשר העכבר יוצא, עוזב אלמנט מסוים, שימושי בעיקר בתיבות טקסט וכדומה.

Keydown - אירוע זה מתרחש כאשר המשתמש מתחיל ללחוץ על מקש כלשהו.

Keyup - אירוע המתרחש כאשר מקש לחוץ משתחרר, בד"כ יבוא בתוכנית שמוגדר שם אירוע **keydown** המתאר מה יקרה בעת לחיצה בלחצן כלשהו במקלדת, והאירוע **keyup** יתאר מה יקרה ברגע שישחררו את הלחיצה.

דוגמא:

```

<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("input").keydown(function(){
        $("input").css("background-color", "yellow");
    });
    $("input").keyup(function(){
        $("input").css("background-color", "pink");
    });
});
</script>
</head>
<body dir="rtl">
ם<input type="text">הכנס ש
</body>
</html>

```

בדוגמא זו כשנמלא ערך בתיבה בעת ההקלדה בעת לחיצה על אחד ממקשי המקלדת יופיע רק בגוון צהוב, בעת שחרור לחצן במקלדת יופיע רקע בגוון וורדרד.

אירועים נוספים:

Focusin() - מתאר מה יקרה כאשר אלמנט מסוים נמצא בפוקוס, שימושי בעיקר בטפסים, כשנכנסים לשדה מסוים בטופס.

Focusout() - מתאר מה יקרה כאשר אלמנט מסוים יוצא ממצב פוקוס, שימושי בעיקר בטפסים, כשעוזבים שדה מסוים בטופס.

המתודה `.on()`

על מנת לקשור אירוע לאובייקט מסוים במסמך ה-`html` נשתמש במתודה `.on`. מתודה זו קושרת אירועים לאובייקטים, בתוך שורת הגדרת מתודה זו יבואו אירועים שונים. לדוגמא:

```
$("p").on("click", function(){  
  
    $(event.currentTarget).hide();  
  
});
```

דוגמא זו מתארת שבעת לחיצה על פסקה בעמוד תתרחש פעולת הסתרה והפסקה תוסתר. ישנם שיטות נוספות לקשירת אירועים, השימושית מבניהם היא השיטה `.on`. קיימות שיטות נוספות `one` ו-`.bind`.

פעולות שניתן להשתמש בהן בפקודות `jQuery`:

`hide()` - הסתרת האלמנט ושחרור מקומו בדף.

`show()` - ההיפך מ-`hide` מציג אלמנט מוסתר.

`toggle()` - להציג ולהעלים, במידה והאלמנט מוסתר, הוא יציג אותו במידה והוא מוצג הוא יסתיר אותו.

`fadeIn()` - להציג אלמנט באפקט דהייה, ניתן להכניס ערך מספרי המציין ערך של מיליסקנד שניות, או את מחרוזת "slow", המציינת שהדהייה תהיה איטית, או את המחרוזת "fast" המציינת שהדהייה תהיה מהירה.

`fadeOut()` - הפוך מ-`fadeIn` מעלים את האלמנט באפקט דהייה, ניתן להכניס לו ערך לציון משך זמן הדהייה, איטי או מהיר.

`fadeToggle()` - מביא את המצב ההפוך באפקט דהייה כשהאלמנט מוסתר יציג באפקט דהייה כשהאלמנט מוצג יסתיר באפקט דהייה.

```
$(document).ready(function(){
```



```
$("#button").on("click", function(){  
  
    $("#div1").fadeToggle();  
  
});  
  
});
```

`slideDown()` - אפקט זה מגדיר לאלמנט לגלוש מלמעלה למטה ולהיות מוצג.

`slideUp()` - אפקט זה מגדיר לאלמנט לגלוש כלפי מעלה ולהעלם.

`slideToggle()` - אפקט זה יציג את המצב ההפוך, אם האובייקט מוסתר, הוא יציג את האובייקט בגלישה מלמעלה למטה, אם האובייקט מוצג, האפקט יגדיר לאלמנט לגלוש כלפי מעלה ולהעלם.

`fadeTo(1, 0.5)` - אפקט זה מעלים אובייקט לשקיפות מסוימת בתחום הסוגריים יש להגדיר שני ערכים, ערך המוצא, וערך היעד.

`Animate()` - להפעלת אנימציה מותאמת אישית לאלמנטים שנבחרו.

לדוגמא:

```
$("#button").on("click", function(){  
  
    $("#box").animate({height: "300px"});  
  
});
```

`delay()` - גורם לעיכוב של הפעולה.

לדוגמא:

```
$("#button").on("click", function(){  
  
    $("#div1").delay("slow").fadeIn();  
    $("#div2").delay("fast").fadeIn();  
  
});
```

`Stop()` - אפקט זה עוצר את האנימציה הפעילה על האלמנט.

לדוגמא:

```
$("#button").on("click", function(){  
  
    $("#div").stop();  
});
```

אירוע זה מגדיר שבעת לחיצה על הלחצן button יוצגו div1 ו-div2 באפקט דהייה, כשהראשון מגיע בהשהיה איטית והשני מגיע בהשהיה מהירה.

הערה: ניתן להגיד לכל הפעולות את קצב הפעולה באמצעות מתן ערכים בתוך הסוגריים של הפעולה, הערכים האפשריים הם "slow", ערך זה מציין שהפעולה תהיה איטית, המחרוזת "fast" המציינת שהפעולה תהיה מהירה, ניתן גם להגדיר ערך מספרי במיליסקנד שניות, ערך ברירת המחדל אם לא נגדיר כלום הוא 400.

תרגיל 113

1. בנו מסמך `html`.
2. הוסיפו במסמך 4 תמונות.
3. קבעו שבמעבר עכבר התמונות תוצגנה או תוסתרנה, בהתאם למצבן (אם התמונה מוצגת תוסתר ולהיפך).

תרגיל 114

1. בנו מסמך `html` עם תמונות של 5 בעלי חיים
 2. קבעו שבעלי החיים יהיו מוסתרים (בעזרת `display:none :css`).
 3. הוסיפו 5 כפתורים
 4. על כל כפתור רשמו שם של בעל חיים אחר
- דוגמא לכפתור (לתזכורת):

```
<button class="lion">אריה</button>
```

5. לחיצה על הכפתור תציג את בעל החיים המצוין על הכפתור באפקט גלישה כלפי מטה.
6. קבעו לכל כפתור השהייה בקצר שונה.

תרגיל 115

1. בנו מסמך `html` חדש.
2. הקלידו בפסקה שיר קצר של כמה מילים.
3. הוסיפו כפתור, רשמו על הכפתור "להצגת השיר".
4. קבעו שהשיר יופיע בלחיצה על הלחצן, באפקט דהייה.
5. קבעו שהשיר יוסתר בעת לחיצה נוספת על הלחצן, באפקט דהייה.

תרגיל 116

1. פתחו את הקובץ פירות.
2. הגדירו לכל לחצן שיבצע את הפעולה שהוא מתאר.

תרגיל 117

1. בנו טופס בסיסי, הכולל תיבת שם פרטי ושם משפחה.

2. הוסיפו לחצן, כתבו עליו את המילים "להצגת הטופס".
3. לחיצה על הלחצן תציג את הטופס, באמצעות אחד מפעולות ההצגה.
4. הוסיפו בטופס לחצן "סגור", לחיצה על הלחצן תסגור את הטופס. (תסתיר אותו).

מניפולציה אחת למספר אובייקטים ושינויי CSS באמצעות jQuery

event.currentTarget

כעת נרצה לתת את אותה מניפולציה למספר אובייקטים מבלי להצמיד לכל אובייקט את המניפולציה מחדש, נוכל להגדיר שבלחיצה על אובייקט מסוים, המניפולציה תחול רק עליו, לדוגמה הגדרנו לכל התמונות class בשם bigImage הצמדנו לתמונות את אירוע הבא:

```
$(document).ready(function(){
    $("bigImage").on("mouseover", function(){
        $("bigImage"). fadeToggle();
    });
    $("bigImage").on("mouseout ", function(){
        $("bigImage"). fadeToggle();
    });
});
```

כאשר נלחץ על אחת התמונות, כל התמונות תעלמנה/תוצגנה, על מנת שהאירוע יחול רק על התמונה עליה אנחנו לוחצים, נעביר לפונקציה את ה-event כך, וכך נוכל להשתמש במאפיין event.currentTarget, המציין שהפעולה תפעל על התמונה עליה לחצו.

```
$("bigImage").on("mouseover", event => {
    $(event.currentTarget).fadeToggle();
});
$("bigImage").on("mouseout ", event => {
    $(event.currentTarget).fadeToggle();
```

```
});
```

תרגיל 118

1. בנו "גלריית תמונות", הגדירו שתמונה שעוברים עליה תקבל אפקט של שקיפות.
2. כאשר עוזבים את התמונה, התמונה תקבל אפקט של אטימות.

תרגיל 119

1. בנו מסמך html חדש.
2. כתבו במסמך 5 פסקאות.
3. כאשר עוברים על כל אחת מהפסקאות, הפסקה תעלם.

פונקציות לביצוע מניפולציות על HTML CSS

ב jQuery ניתן לבצע שינויים על ה- HTML וה- CSS של המסמך באמצעות הפונקציה `css()`

פונקציה זו מקבלת זוגות של שמות מאפיינים וערכים, כשם שהם נכתבים ב-CSS וכך ניתן לשנות מספר מאפיינים של אלמנט במסמך.

לדוגמא:

```
$(document).ready(function(){  
  
  $("button").click(function(){  
  
    $("p").css("background-color", "yellow");  
  
  });  
  
});
```

בדוגמא זו קבענו שבעת לחיצה על הלחצן `button`, כל הפסקאות בדף תקבלנה צבע רקע צהוב.

תרגיל 120

1. פתחו מסמך `html`.

2. כתבו במסמך 5 פסקאות שונות.

3. קבעו שבעת לחיצה על הפסקה, צבע הטקסט של הפסקה יקבל צבע אדום.

אך כשם שלא מומלץ לשלב CSS בגוף המסמך מסיבות של עקביות וסדר, כך גם ב-jquery מומלץ שלא להכניס הגדרות CSS, אלא הפרקטיקה העדיפה היא השארת הגדרות ה-CSS בקובץ ה-CSS ושליטה במסמך באמצעות ה-jquery על ידי הוספת והסרת class מסויים מאלמנט.

הפונקציה `addClass()`

הוספת קלאס תבצע באמצעות הפונקציה `addClass()` כששם הקלאס מוגדר בתוך גרשיים תחום בסוגריים כך:

```
$("#button").on("click", function(){  
  
    $("#p").addClass("example");  
});
```

ל-class `example` נגדיר הגדרות CSS בקובץ ה-CSS כשם שמגדירים ערכים לכל class. כאשר נרצה שה-class שהוספנו יסיים את השפעתו נשתמש בפונקציה `removeClass()`

הפונקציה `removeClass()`

הפונקציה `removeClass()` מסירה מהאלמנט השפעה של class קיים, כששם הקלאס שאותו נסיר מוגדר בתוך גרשיים תחום בסוגריים כך:

```
$("#button").on("click", function(){  
  
    $("#p").removeClass("example ");  
});
```

הפונקציה `toggleClass()`

הפונקציה `toggleClass()` מסירה/מוסיפה לאלמנט class מסוים המוגדר בין גרשיים, תחום בסוגריים, אם ה-class קיים לאלמנט הפונקציה מסירה את ה-class, אם הוא אינו קיים, הפונקציה מוסיפה אותו.

```
$(document).ready(function(){  
  
  $("button").click(function(){  
    $("h1, h2, p").toggleClass("Isdisplayed");  
  });  
});
```

בדוגמה זו הפונקציה `toggleClass` מוסיפה את ה-class במידה והוא לא קיים, אם הוא קיים היא מסירה אותו.

תרגיל 121

1. צרו מסמך html חדש
2. הגדירו לכל הפסקאות צבע אדום
3. הוסיפו לחצן, שבלחיצה אחת מוסר הקלאס המגדיר לפסקאות את הצבע האדום.
4. הוסיפו לחצן נוסף, כתבו עליו "כתום".
5. הגדירו class בשם orange.
6. קבעו שבעת לחיצה על הלחצן "כתום", כל הפסקאות הופכות צבען לצבע כתום.
7. הוסיפו לחצן נוסף כתבו עליו "הסתר טקסט", קבעו שבעת לחיצה על הלחצן, כל הפסקאות נעלמות בעזרת הוספת `css` `(display: none)`.

תרגיל 122

1. פתחו מסמך html חדש
2. קבעו שכל הפסקאות תקבלנה צבע כחול בלחיצה על אחת מהפסקאות, בלחיצה נוספת הפסקאות תחזורנה לצבען המקורי. (השתמשו בפונקציה `(toggleClass())`).

תרגיל 123 (אתגר)

1. הכינו משחק חשבון, במשחק יוצגו 9 משבצות.
2. בכל משבצת כתבו תרגיל חשבוני.
3. בעת לחיצה על התרגיל, יפתח פופאפ להזנת תשובה לתרגיל. `(Window.prompt)`.
4. הכינו שתי הגדרות `css`:
 - a. לקלאס ששמו `smile`, הקלאס יקבל תכונת רקע תמונה כך:

```
smile {background-image: url(nosmile.jpg);}
```
 - b. לקלאס ששמו `nosmile`, הקלאס יקבל תכונת רקע תמונה כך:

```
nosmile {background-image: url(nosmile.jpg);}
```
5. אם התשובה נכונה המשבצת תתמלא בתמונת רקע של סמיילי, אחרת המשבצת תתמלא בתמונת סמיילי עצוב.

6. דאגו שבכל תרגיל יוכלו ללחוץ פעם בודדת (one במקום on גורם לפעולה להיות מוגבלת לפעם בודדת).

הפונקציה attr()

ניתן לשלוט במאפיין של אלמנט HTML באמצעות הפונקציה attr(), הפונקציה attr מקבלת שני ערכים, שם המאפיין וערך לשינוי, לדוגמא אם נרצה לשנות מאפיין של קישור

```
$(document).ready(function(){  
  
  $("button").click(function(){  
  
    $("a").attr("href", "https://www.gov.il/");  
  
  });  
  
});
```

תרגיל 124

1. צרו מסמך html חדש הוסיפו תמונה כרצונכם.
2. הוסיפו לחצן כתבו עליו "שנה תמונה".
3. בעת לחיצה על הלחצן, התמונה תשתנה לתמונה שונה. (באמצעות attr המשנה את המאפיין src).
4. שנו את ה-title ואת ה-alt בהתאם לתמונה החדשה.
5. בעת מעבר על התמונה, התמונה תשתנה לתמונה המקורית, הן title והן alt ישתנה בהתאם.

הפונקציות `html()` ו-`text()`הפונקציה `html()`

ניתן לקבל את תוכן ה HTML של אלמנט מסוים באמצעות הפונקציה `Html()`, ניתן גם באמצעות פונקציה זו לשנות את תוכנו של ה-`html`, ניתן לשנות תוכן אלמנט במסמך באמצעות שליחת תוכן חדש בפונקציה.

לדוגמא:

```
$(document).ready(function(){
    $("button").click(function(){
        $("p").html("<b>יום נעים ומוצלח</b>");
    });
});
```

בדוגמא זו בלחיצה על הלחצן, כל הפסקאות במסמך מקבלות את הטקס יום נעים ומוצלח, כשהמילה מוצלח, מודגשת, כיוון ששלחנו אותה בתוך תגית עיצובית, התגית `b` שמשפיעה על עיצובה ומדגישה אותה, התוכן המקורי של הפסקאות ימחק.

באמצעות הפונקציה `html` ניתן גם לקבל את תוכן האלמנט לדוגמא:

```
alert( $("a").html() );
```

דוגמא זו תדפיס לחלונית פופאפ את תוכן האלמנט `a`, כלומר את הטקסט המוצג ב-`html`.

הוספת תוכן תוך שמירה על התוכן המקורי

ניתן להוסיף תוכן למסמך באמצעות הפונקציה `html` כשהתוכן המקורי ישמר ולא יידרס. כיצד?

נגדיר משתנה בשם `string` (ניתן לתת כל שם שהוא).

נכניס בתוכו את ערך התוכן של הפסקה אליה נרצה לצרף תוכן חדש באמצעות פעולת השמה:

```
let string = $("#add").html();
```

נרשרר למשתנה string את תוכן ה-html החדש:

```
string += "<p>תוכן חדש</p><span>ממש לאלמנט</span></p>"
```

כעת נוצרה לנו מחרוזת חדשה המכילה את התוכן הקודם בצירוף התוכן החדש, נכניס את המשתנה המחזיק את המחרוזת לפונקציה html:

```
$("#add").html(string);
```

בצורה כזו קבלנו תוכן חדש בתוספת על הישן.

הפונקציה text()

הפונקציה text דומה מאד לפונקציה html, אך בשונה מפונקציית html כאשר נשלח לה ערכים עם תגיות html התגיות תודפסנה על המסך ולא תתקבלנה כתגיות המשפיעות לעיצוב הטקסט הנשלח.

לדוגמא:

```
$("#text").text("יום נעים<b>!ומוצלח</b>");
```

```
$("#p").html("יום נעים<b>!ומוצלח</b>");
```

הפלט שיוצג על המסך כתוצאה משתי השורות הנ"ל הוא כזה:



בשורה הראשונה נוכל לראות שהודפסו על המסך כתוצאה מהפונקציה text גם תגיות ה-html, לעומת זאת אותו קוד הנשלח באמצעות הפונקציה html, החזיר תוצאת פלט המוצגת בהשפעת התגיות שנשלחו בפונקציה (התגית b).

הפונקציה val()

הפונקציה Val() נועדה לקבל (ולשים) ערך לשדות בטפסים.

```
$(document).ready(function(){  
  
    $("button").click(function(){  
  
        $("input:text").val("תקוה כהן");  
  
        alert($("input:text").val());  
  
    });  
  
});
```

בדוגמא זו אנחנו רואים יישום של שני התפקידים של הפונקציה val, הכנסת ערך טקסט לשדה הקלט, ובנוסף הדפסת ערך שדה הקלט לחלון alert, (הערה: כיוון שהשדה ריק אם היינו הופכים את סדר ההוראות, פקודת ה-alert הייתה מדפיסה פופאפ ריק, כיוון שהשדה היה לפני כן עם ערך ריק והכנסנו באמצעות הפונקציה val.

תרגיל 125

1. בנו מסמך html המכיל 5 פסקאות.
2. לחיצה על כל אחת מהפסקאות תשנה את הטקסט שלה למילה "נמחק".

תרגיל 126

1. בנו טופס בסיסי, המכיל שדה להזנת שם פרטי ושם משפחה.
2. הוסיפו בטופס לחצן, כתבו עליו "מלא", לחיצה על הלחצן תמלא את שדות הטופס בשמכם ושם משפחתכם.

תרגיל 127

1. בנו טופס המכיל שם מוצר, מחיר, כמות, לחצן חשב ותוצאה.
2. לחיצה על הלחצן חשב תדפיס לשדה תוצאה את תוצאת החישוב.

1. בנו מחשבון בסיסי.
2. במחשבון לחצני מספרים מ-0 עד 9, שדה תרגיל, לחצן חיבור, לחצן חיסור, לחצן כפל, לחצן חילוק, ושדה תוצאה.
3. לחיצה על כל אחד מלחצני החישוב (חיבור חיסור כפל וחילוק), תדפיס לשדה תוצאה את תוצאת התרגיל עד כה.

ניווט בין אלמנטים

לעיתים נרצה להתייחס לאובייקטים אשר אין לנו מידע כמו class שם אלמנט, או id המתאר את האובייקט באופן פשוט, לדוגמא אם נרצה לגשת לכל הבנים של אובייקט מסוים, או לאבא של אובייקט מסוים, באחד מהשיעורים העוסקים בעיצוב ב-CSS למדנו לבנות סלקטורים ספציפיים יותר המאפשרים לנו לגשת למגוון של אובייקטים, בדומה לכך קיימות פונקציות ניווט המאפשרות לנווט בין האובייקטים במסמך ה-HTML ולגשת לאלמנטים שונים במסמך.

פונקציית הניווט children()

פונקציה זו מחזירה את כל הילדים הישירים של אלמנט שנבחר.

לדוגמא במבנה הבא:

```
<body class="descendants">body (סבא רבא)
  <div style="width:500px;">div (סבא)
    <ul>ul (הורה ישיר)
      <li>li (ילד)
        <span>span (נכד)</span>
      </li>
      <li>li (ילד)
        <span>span (נכד)</span>
      </li>
    </ul>
  </div>
</body>
```

ההוראה:

```
$("#ul").children().css({"color": "orange", "border": "5px solid red"});
תוסיף מסגרת לאלמנטים li שהם הבנים הישיר של ul.
```


ניתן לשלוח לפונקציה זו פרמטרים, לדוגמא אם לאובייקט יש מספר בנים ישירים ממספר סוגים, נוכל לשלוח את סוג האלמנט אותו נרצה לבחור.

פונקציית הניווט `siblings()`

פונקציה זו מחזירה את כל האחים מאותה רמה, אחים הם אלמנטים מקבילים ברמת ה-`.html`

לדוגמא במבנה הבא:

```
<div>
  <ul>רשימה
    <li>פריט ראשון</li>
    <li>פריט שני</li>
    <li class="one">פריט שלישי</li>
    <li>פריט רביעי</li>
    <li>פריט חמישי</li>
  </ul>
</div>
```

כל הפריטים (ה-`li`ים) ברשימה הם אחים.

```
$("#li.one").siblings().css({"color": "red", "border": "2px solid red"});
```

אם בחרנו את האחים של האובייקט עם ה-`one class`, יבחרו כל פריטי הרשימה מלבד ה-`li` עם ה-`one class`, שהוא אינו אח של עצמו.

פונקציית הניווט `Parent()`

מחזירה את האבא הראשון של האלמנט.

לדוגמא:

```
$(document).ready(function(){
  $("#span").parent().css({"color": "red", "border": "2px solid red"});
});
```

מבנה ה-`html`:

```

<body class="ancestors">body (great-great-grandparent)

<div style="width:500px;">div (great-grandparent)

<ul>ul (grandparent)

  <li>li (direct parent)

    <span>span</span>

  </li>

</ul>

</div>

</body>

```

הוראה זו תתן מסגרת בצבע אדום לאלמנט li, לעמות זאת פונקציית הניווט parents() תבחר את כל האלמנטים ההורים של האובייקט span כך שבשימוש ב parents() כל האלמנטים מלבד ה-span יקבלו מסגרת אדומה.

פונקציית הניווט Children()

מחזירה את כל הילדים הישירים של אובייקט מסויים.

לדוגמא:

```

$(document).ready(function(){
  $("ul").children().css({"color": "red", "border": "2px solid red"});
});

```

הוראה זו תתן מסגרת בצבע אדום לאלמנט li, שהוא בן ישיר של ה-ul, מבנה ה-html הובא למעלה.

פונקציית הניווט Closest()

מחזירה את האבא הראשון לאלמנט שנבחר, האבא שהכי קרוב אליו, פונקציה זו מקבלת כערך את שם האלמנט לחיפוש, לדוגמא:

```
$(document).ready(function(){  
  $("span").closest("ul").css({"color": "red", "border": "2px solid red"});  
});
```

בדוגמא זו האובייקט ul (שאותו העברנו בפונקציה) הקרוב ביותר ל-li זה שעומד מעליו בסמוך אליו, יקבל את המסגרת האדומה.

פונקציית הניווט Find()

מחזירה אלמנטים צאצאים של האובייקט שנבחר, ניתן להעביר לה כערך את שם האלמנט אותו אנחנו מחפשים.

פונקציית הניווט Next()

מחזירה את האלמנט הבא בתור במבנה ה-html, מחזירה את האח הבא, אלמנט העומד באותה רמה.

לדוגמא:

```
$(document).ready(function(){  
  $("li#new ").next().css({"color": "red", "border": "2px solid red"});  
});
```

```
<div style="width:500px;" class="siblings">
```

```
<ul>ul
```

```
<li>li </li>
```

```
<li>li</li>
```

```

<li id="new">li </li>

<li>li </li>

<li>li </li>

</ul>

</div>

```

בדוגמא זו נקבל את האלמנט הבא אחרי ה-li עם ה-id new.

פונקציית הניווט Prev()

הפוך מהפונקציה next, מחזירה את האח הקודם של האלמנט. קיימות פונקציות ניווט נוספות, הכרנו את העיקריים והחשובים מבניהם.

תרגיל 129

1. צרו מסמך html חדש.
2. הוסיפו במסמך רשימה הכוללת 9 ערכים.
3. הוסיפו לאחד מערכי הרשימה id בשם brother.
4. קבעו שכל האחים של האלמנט # brother יקבלו צבע רקע אפור.
5. קבעו שהאלמנט בן הראשון של ה-ul יקבל צבע טקסט כתום.
6. הוסיפו רשימה נוספת, הכוללת 4 ערכים.
7. קבעו באמצעות הפונקציה Next() שהרשימה השנייה תקבל צבע רקע צהוב.

תרגיל 130

1. העתיקו את מבנה ה-html הבא:

```

<body class="ancestors">body (סבא רבא רבא)
  <div style="width:500px;">div (סבא רבא)
    <ul>ul (סבא)
      <li>li (הורה ישיר לספאן)
        <span>span</span>

```

```

        </li>
        <li>li (הורה ישיר לספאן)
            <span>span</span>
        </li>
    </ul>
</div>
</body>

```

2. קבעו באמצעות הפונקציה `children()` שכל הפרטים ברשימה יקבלו צבע טקסט כחול.

3. קבעו באמצעות הפונקציה `find()` שהאלמנט `span` יקבל צבע רק אפור.

chaining

ב jQuery ניתן להפעיל מספר פונקציות על אותו אובייקט מבלי לקרוא לו מחדש על ידי שרשור של הפונקציות בזו אחר זו, לדוגמא:

```

$(document).ready(function(){
    $("button").click(function(){
        $("p").css("color","orange").slideUp(2000).slideDown(2000);
    });
});

```

הפונקציות יופעלו בזו אחר זו.

תרגיל סיכום

1. בנו לעצמכם אתר.
2. האתר יכלול 3 עמודים מקושרים.
3. עמוד דף הבית:
 - a. הדף יכלול תמונה שלכם או כל תמונה אחרת המאפיינת אתכם או את המקצוע שלכם, כתבו מעט על המקצוע שאתם עוסקים בו, התחביב שלכם, ומה אתם אוהבים לעשות.
4. הוסיפו עמוד הנקרא גלריה:
 - a. העמוד יכלול גלריית תמונות שלכם או של עבודות שעשיתם.
 - b. הוסיפו מרווחים יפים בין התמונות.
 - c. קבעו לכל תמונה גבול, קבעו שבעת מעבר על אחת מהתמונות, התמונה תקבל אפקט דהייה (fadeIn).
5. עמוד צור קשר:
 - a. עמוד זה יכלול פרטי התקשרות: טלפון (טלפון לחיץ להתקשרות), כתובת מייל (לחיצה לשליחת אימייל), ומיקום גאוגרפי.
 - b. טופס ליצירת קשר הכולל את השדות:
 - c. פנייה בנושא, תיבה נפתחת (הזמנת עבודה, הצעה לשיפור, פנייה חוזרת, אחר).
 - d. תוכן הפנייה (אזור טקסט המוגבל ל800 תווים).
 - e. שם, כתובת, אימייל.
 - f. בכל השדות יופיע placeholder.
 - g. בדף הבית שלבו אנימציה מעניינת.
6. הכינו תפריט, התפריט ימוקם בכל העמודים בחלקו העליון של הדף התפריט יקשר בין חלקי האתר.
7. הקפידו על בנייה נכונה וכתיבת כל תוכן בתגית המתאימה לו, שלבו תמונות במקומות שניתן.