

---

# HTML5

---

## המדריך לבניית אתרים ולמערכות Web

הדור הבא

מהדורה שלישית  
(עדכון 2018)

יונית רושו



יועץ מקצועי: **זהר עמיהוד**

עריכה ועיצוב: **שרה עמיהוד, יצחק עמיהוד**

עיצוב עטיפה: **אופיר ליברמן**

## שמות מסחריים

שמות המוצרים והשירותים המוזכרים בספר הינם שמות מסחריים רשומים של החברות שלהם. הוצאת הוד-עמי והמחברת עשו כמיטב יכולתם למסור מידע אודות השמות המסחריים המוזכרים בספר זה ולציין את שמות החברות, המוצרים והשירותים. שמות מסחריים רשומים ( registered trademarks) המוזכרים בספר צוינו בהתאמה.

## הודעה

ספר זה מיועד לתת מידע אודות מוצרים שונים. נעשו מאמצים רבים לגרום לכך שהספר יהיה שלם ואמין ככל שניתן, אך אין משתמעת מכך כל אחריות שהיא.

המידע ניתן "כמות שהוא" ("as is"). הוצאת הוד-עמי והמחברת אינם אחראים כלפי יחיד או ארגון עבור כל אובדן או נזק אשר ייגרם, אם ייגרם, מהמידע שבספר זה, או מהתקליטור/הדיסקט (אם יצורף), או מאתר האינטרנט המלווה את הספר.

אין לעשות שימוש מסחרי ו/או להעתיק, לשכפל, לצלם, לתרגם, להקליט, לשדר, לקלוט ו/או לאחסן במאגר מידע בכל דרך ו/או אמצעי מכני, דיגיטלי, אופטי, מגנטי ו/או אחר - בחלק כלשהו מן המידע ו/או התמונות ו/או האזורים ו/או כל תוכן אחר הכלולים ו/או שצורפו לספר זה, בין אם לשימוש פנימי או לשימוש מסחרי. כל שימוש החורג מציטוט קטעים קצרים במסגרת של ביקורת ספרותית אסור בהחלט, אלא ברשות מפורשת בכתב מהמוציא לאור.

**לשם שטף הקריאה כתוב ספר זה בלשון זכר בלבד. ספר זה מיועד לגברים ונשים כאחד ואין בכוונתנו להפלות או לפגוע בציבור המשתמשים/ות.**

**(C) כל הזכויות שמורות**

**הוצאת הוד-עמי בע"מ**

טלפון: 09-9564716

[www.hod-ami.co.il](http://www.hod-ami.co.il)

[info@hod-ami.co.il](mailto:info@hod-ami.co.il)

הודפס בישראל 11/2014, 2/2016, 5/2016, 3/2018

מסת"ב 978-965-361-424-6 ISBN

**הספר מוקדש באהבה לבעלי היקר אורן ולילדים שלי -  
שקד, דניאל וירין רושו, על כך שנתתם לי את הכוח  
ותמכתם לכל אורך הדרך**



# תוכן עניינים מקוצר

---

19	הקדמה
21	פרק 1: מבוא ומושגים באינטרנט
25	פרק 2: יסודות ועקרונות שפת <b>HTML(5)</b>
87	פרק 3: תגיות המבנה החדשות
105	פרק 4: הגדרות עיצוב על ידי <b>CSS</b>
147	פרק 5: מודל הקופסה ופריסות
183	פרק 6: קוד מינימליסטי ובעל משמעות
189	פרק 7: טפסים ב- <b>HTML5</b>
225	פרק 8: נושאים מתקדמים <b>CSS3</b>
265	פרק 9: <b>Javascript</b> ו- <b>Jquery</b>
299	פרק 10: שילוב מדיה והאובייקט <b>Canvas</b>
333	פרק 11: נושאים מתקדמים <b>API</b>
349	פרק 12: מובייל ורספונסיביות
385	אינדקס

# תוכן עניינים

---

19	הקדמה
19	מבנה הספר
20	קהל היעד
20	קטעי הקוד
20	על יונית רושו
21	<b>פרק 1: מבוא ומושגים באינטרנט</b>
22	כתובות של מחשבים ברשת האינטרנט
22	טכנולוגיית שרת לקוח
22	הדפדפן
23	פרוטוקול HTTP
23	פרוטוקול FTP
23	אחסון האתר
24	דומיין
24	תת-דומיין
24	HTML (HyperText Markup Language)
24	סיכום
25	<b>פרק 2: יסודות ועקרונות שפת HTML(5)</b>
26	מה זה HTML5?
26	במה שונה HTML5 מהגרסאות הקודמות?
27	תמיכת דפדפנים
27	דף HTML5 הראשון שלך
31	תגיות, מאפיינים ואלמנטים בסיסיים של השפה
31	התגית: אבן היסוד של כל דף HTML
32	מבנה סטנדרטי של מסמך HTML5: תגיות היסוד
32	DOCTYPE: הצהרה על HTML5
32	התגית <html>
33	התגית <head>
33	התגית <title>
33	התגית <body>

34	.....<section>	התגית
35	.....	תגיות נפוצות נוספות להגדרת אזורים בדף
35	.....	עברית בדפי אינטרנט
36	.....	תוכן טקסטואלי
36	.....	הכרת המושג כותרות
37	.....	תגיות הכותרת
39	.....<p>	תגית הפסקה
39	.....	תגיות בסיסיות לעיצוב הטקסט
39	.....	חוקיות הפתיחה והסגירה של תגית בתוך תגית
40	.....	התוכן שבין התגיות
40	.....	להיכן נעלמו הרווחים בין המילים ובין השורות?
41	.....&nbsp;	רווח קשיח
42	..... 	מעבר שורה
42	.....	שילוב תווים מיוחדים בתוכן שבין התגיות - ASCII Characters
43	.....	מאפיינים לתגיות - Attributes
44	.....	סיכום: התקניות הנדרשת במסמכי HTML5
47	.....	תרגיל 1
48	.....	קישורים
51	.....a	תגית הקישור
51	.....	המאפיין href
52	.....	המאפיין target
52	.....	המאפיין rel
52	.....	המאפיין media
53	.....mailto	קישור לדואר אלקטרוני
54	.....	קישור למסמך Word שנמצא באותה תיקייה
54	.....	קישור לדף אחר באתר שלי שנמצא באותה התיקייה
54	.....	קישור פנימי בתוך דף Anchors-
56	.....	התגית base
57	.....	תרגיל 2
58	.....	צבעים ורקע
58	.....	רשימות ותבליטים
61	.....	רשימות תבליטים לא ממוספרות
62	.....	רשימות תבליטים ממוספרות
62	.....	רשימת הגדרות - Definition list
63	.....	מאפיינים (attributes) של רשימות

63	המאפיין reversed
63	המאפיין start
63	שינוי מראה הרשימה
66	שילוב תמונות בדף
71	מידע כללי
72	JPG
72	GIF
72	קבצי GIF ANIMATION
72	PNG
72	התגית img
74	סידור סביבת העבודה - התיקיה images
74	נתיבי קבצי תמונות
74	נתיב יחסי (Relative path)
75	נתיב מוחלט (Absolute path)
75	שילוב תמונה מתיקיית התמונות באתר
75	תמונה כקישור
78	הוספת תמונת רקע
80	תרגיל 3
81	טבלאות
82	תאים ריקים בטבלה
83	מרווחים בין תאים ובתוך התאים
85	סיכום
<b>87</b>	<b>פרק 3: תגיות המבנה החדשות</b>
88	תגיות חדשות ב-HTML5 ליצירת מבניות בעלת משמעות
88	התגית <header>
88	התגית <footer>
88	התגית <section>
89	התגית <article>
90	התגית <nav>
92	התגית <main>
92	התגית <aside>
93	אזורים יעודיים
95	תגיות חדשות נוספות
95	התגית <figure>



95	.....<figcaption>	התגית
96	.....<mark>	התגית
97	.....HTML5	מודל התוכן של
97	..... Inline	אלמנטים מסוג
99	..... DOM	ומודל התוכן החדש: 7 הקטגוריות הראשיות
100	..... Metadata	
100	..... Flow	
100	..... Heading	
101	..... Phrasing	
101	..... Embedded	
101	..... Interactive	
101	..... Sectioning	
101	..... (Outline algorithm)	אלגוריתם קווי המתאר של התוכן
102	..... HTML5	תמיכה בדפדפנים שאינם תומכים ב-
103	..... 4	תרגיל
104	.....	סיכום
<b>105</b>	..... <b>CSS</b>	<b>פרק 4: הגדרות עיצוב על ידי</b>
105	..... ?CSS	מה זה ?
109	..... CSS	גרסאות של
109	..... ?CSS	היכן כותבים את הגדרות ?
110	..... Inline	הגדרת העיצוב ישירות בתגית HTML: שיטת
112	..... Inline	הגדרות צבע בשיטת ובאופן כללי.
113	..... Inline	הבעיה בשיטת
113	..... Internal	הפתרון: שיטת
114	..... <style>	הגדרת העיצוב בראש המסמך באופן גורף: התגית
115	..... <style>	הבעיה בשיטת תגית
116	.....	הפתרון
116	..... CSS	השיטה השלישית והנבחרת: קובץ
117	.....	מה עושה הדפדפן?
117	..... CSS	דוגמאות
122	.....	הגדרות CSS שימושיות לפי נושאים
122	.....	רקעים
122	..... background-color	- צבע רקע
122	..... background-image	- תמונת רקע

123	background-position - מיקום תמונת רקע
123	background-repeat - שכפול תמונת רקע
123	מסגרות
124	border-width - עובי מסגרת
124	border-style - סגנון מסגרת
125	border-color - צבע מסגרת
125	טיפוגרפיה: הגדרות עיצוב עבור טקסטים וגופנים
125	שימוש בטקסט 'חי'
126	Web-safe fonts - גופן מערכת
126	font-size, font-family, color - צבע הגופן, משפחה וגודל
127	font-style - סגנון הגופן
127	font-weight - עובי הגופן
127	direction - כיוון הטקסט
128	letter-spacing - רווח בין אותיות
128	line-height - גובה שורה
128	text-align - יישור טקסט
128	vertical-align - יישור אנכי
129	word-spacing - רווח בין מילים
129	@font-face - צורת הגופן
129	עיצוב קישורים
129	text-decoration - קישוט הטקסט
129	lists - עיצוב רשימות
130	שוליים
130	margin - שוליים חיצוניים
130	padding - שוליים פנימיים
131	גדלים: רוחב וגובה
131	איחוד הגדרות למספר תגיות
131	תרגיל 5 שלב 1
132	הגדרות עיצוב שאינו גורף: Class
134	המשך תרגיל 5 - שלב 2
134	סדרי עדיפויות של הגדרות עיצוב
135	כוכבית
136	שימוש בשם, המאפיין id
137	אזורים יעודיים בדף אינטרנט או ביישום
137	שימוש בשתי מחלקות (classes) לאותו אלמנט

137	קישורים אינטראקטיביים
137	Hover - שינוי עיצוב במעבר עכבר מעל קישור
138	Visited - שינוי עיצוב של קישור לאחר ביקור בו
138	יעילות הכתיבה של מסמכי CSS
138	שימוש ב-CLASS במידה
139	עמידה בשני תנאים: הגדרת סוג תגית וגם שימוש ב-class
140	היררכיה ב-CSS, או קינון תגיות
141	ההבדל בין 'צאצאים' לבין 'ילדים' מדור ראשון
143	תגית שעוקבת אחרי תגית אחרת
143	תגיות שנמצאות לאחר תגית מסוימת
144	תגיות בעלות מאפיין מסוים
145	תגיות עם מאפיין בעל ערך מסוים
146	שימוש בהערות
146	סיכום
<b>147</b>	<b>פרק 5: מודל הקופסה ופריסות</b>
148	מבנה ופריסת הדף
149	מודל הקופסה - Box model
152	איחוד שוליים אנכיים - Margin collapse
152	תגית מתחת לתגית
154	הכלת תגית בתוך תגית אחרת
155	איחוד שוליים בתוך אותו האלמנט
156	הגדרות תצוגת Inline ותצוגת Block
156	אלמנטים עם תצוגת Block
157	אלמנטים עם תצוגת Inline
158	מיקום אלמנטים
158	זרימה רגילה של הדף - Normal flow
159	מיקום יחסי - Relative positioning
160	מיקום מוחלט - Absolute positioning
164	דוגמה שימושית
165	מיקום מקובע - Fixed positioning
165	אלמנטים "צפים" - Floating
167	מסך צר של מכשיר נייד
168	תג הפסקת ציפה - Clear
171	מרכזו אופקי

171	מרכז באמצעות שוליים אוטומטיים
172	מרכז אופקי באתרים ברשת
175	מרכז של טקסט בתוך אלמנטים לעומת מרכז של אלמנטים
177	דפדפנים ישנים למישהו?
177	תרגיל 6
181	סיכום
<b>183</b>	<b>פרק 6: קוד מינימליסטי ובעל משמעות</b>
183	למה הכוונה קוד בעל משמעות?
186	הפרדת התוכן והמשמעות מהתצוגה
186	מינימליזם בקוד
188	סיכום
<b>189</b>	<b>פרק 7: טפסים ב-HTML5</b>
190	מהו הטופס ומהי מטרתו?
191	טכנולוגיה ועיצוב יד ביד
193	החשיבות של תכנון ועיצוב הטפסים
197	השפעת התפתחות העיצוב והטכנולוגיה על טפסים ב-HTML5
197	לאן נשלחים הנתונים מהטופס, וכיצד
198	שרת מקומי - Local Server
201	סדנת עבודה
201	התגית <form> ומבנה טופס
203	מבנה הטופס
203	התגית <label>
204	הפקדים
205	הפקדים הבסיסיים: פקדי תיבת טקסט ושליחת הנתונים input
207	פקדי קלט נוספים עבור טקסטים ומספרים
212	פקד אזור טקסט, Textarea
213	פקד הלחצן - button
214	פקדי בחירה
214	תיבת סימון - check box
215	פקדי רדיו - radio buttons
216	רשימה נגללת - select
218	פקד להעלאת קובץ - file upload

219	עיצוב הפקדים והטופס בכללותו
219	פקד "שלח" (submit) מעוצב עם תמונה
219	התגיות <fieldset> ו- <legend>
220	התגית <details>
221	ניהול המידע
221	התגית <datalist>
221	התגית <datagrid>
222	הקוד לבניית הטופס
224	סיכום
<b>225</b>	<b>פרק 8: נושאים מתקדמים CSS3</b>
225	תמונות רקע, חזרה ודגשים
226	הגדרת תמונת רקע עבור מעברי צבע
227	מיקום תמונת רקע
228	מיקום על פי פיקסלים
229	מיקום באחוזים
230	עיצוב קישורים מתקדם
230	קו תחתון בעיצוב אישי
231	טכניקות להקטנת מספר המחלקות (classes)
231	הדגשת קישורים מסוגים שונים
232	תפיסת התחלת מחרוזת הערך
233	תפיסת סוף מחרוזת הערך
234	מעברי עכבר יעילים מעל קישורים
234	מעברי עכבר פשוטים
235	מעברי עכבר בשימוש תמונה אחת - pixy-style rollovers
236	תפריט אופקי
238	טכניקות חדשות ב-CSS3
238	הוספת צללית לטקסט חי
239	עיצוב עמודות
242	שקיפות - Opacity
242	מעברי צבע - gradients
244	ריבוי תמונות רקע
245	מסגרות לתגיות
246	פינות מעוגלות
247	צל לתגיות "קופסה"

248	מסגרות מעוצבות
248	יצירת לשוניות (tabs) עם פינות מעוגלות
250	CSS transform
251	אנימציות ב-CSS3
252	שינוי מצב
253	שינוי מיקום
255	טיפוגרפיה: עיצוב גופנים
258	הטמעת גופנים
260	זכויות יוצרים
260	ייבוא משפחת גופנים, לעומת הטמעת גופן כקובץ
260	ייבוא משפחת גופנים
262	תמיכה בדפדפן Internet Explorer
263	דוגמה שלמה
263	סיכום
<b>265</b>	<b>פרק 9: Jquery ו-Javascript</b>
266	מהי Javascript?
266	התפקיד של Javascript ברף אינטרנטי
266	דף Javascript הראשון שלך
267	כיצד פועלת השפה Javascript?
268	תחביר השפה Syntax
269	אירועים, פונקציות, אובייקטים ומשתנים
270	מודל אובייקט המסמך - DOM
271	האובייקט החשוב - Document
271	תחביר הנקודה
271	קריאת נתונים והשמת נתונים
272	סדר קריאת הקוד על ידי הדפדפן
274	משתנים - variables
274	הגדרה והצהרה של משתנים
275	ערכי משתנים
275	שרשור מספרים למחרוזות
275	התניות והשמות
276	ריבוי התניות
277	שיטות בדיקה
277	הוראת התנאי if

277	ההוראה this - מי אני?
278	פונקציות מובנות שימושיות
278	getElementById(""); הפונקציה המובנית
278	substring הפונקציה המובנית
278	indexOf הפונקציה המובנית
279	פונקציות ואירועים
279	החלפת קובץ התמונה בלחיצה עליה
279	שלב ראשון - תמונה אחת
280	שלב שני - שתי תמונות
281	החלפת תמונה גדולה בהתאם למעבר עכבר מעל תמונה קטנה
283	שינוי טקסט בדף
283	האובייקט Style
284	הצגה והסתרה של שכבה בלחיצה על אובייקט בדף
285	אופן הגדרת פונקציות custom-made (נושא מתקדם)
287	כיווץ - מיניפיקציה, minification
287	ספריית Javascript (Javascript libraries)
287	הספרייה JQuery
288	התקנת הרכיב המרכזי להרצת JQuery
288	כתיבת הקוד הראשון שלך ב-JQuery
289	האירוע document.ready
289	סימן הדולר
290	מציאת אובייקטים על המסך
292	שרשרת הוראות - Chaining
292	שינוי תוכן בעמוד
293	שינוי ערכים של מאפיינים
293	מניפולציות על CSS
294	אירועים
295	הופעה איטית והיעלמות איטית של אובייקטים - Fade
297	שימוש ב-JQuery קיים מהרשת
297	קטעי CSS מוכנים ומסגרות קוד CSS Snippets, Frameworks
298	סיכום
<b>299</b>	<b>פרק 10: שילוב מדיה והאובייקט Canvas</b>
299	שילוב סרטונים ב-HTML5
302	שליטה על אובייקט video באמצעות Javascript

303	שילוב שמע ב-HTML5
303	תגית audio ב-HTML5
304	התגית Canvas
304	הכרת התגית, דפדפנים תומכים, מטרה
305	הוספת התגית למסמך
306	הגדרת גודל הקנבס
307	כתיבת הקוד הראשון בקנבס: hello world
308	תכנות בסיסי של התגית <canvas>
308	מערכת הקואורדינטות
309	המאפיינים המרכזיים של canvas
309	הפונקציות המרכזיות של canvas
309	זיהוי תמיכה בקנבס
310	תכנות בתוך הקוד שתומך בקנבס
311	מהו הציור על הקנבס?
311	אפשרויות הצביעה ב-API ב-2D
313	ציור מלבנים
314	ציור קווים
315	ציור משולשים על הקנבס
317	ציור טקסטים
317	צורות מורכבות: קשתות ונתיבים
319	ציור עיגולים על משטח העבודה canvas
320	עקומות
320	הכרת המצב (state) של משטח העבודה
323	יצירת צל
325	הצגת תמונות בקנבס
325	מסכות - clipping paths
327	טרנספורמציות - transformations
327	שינוי גודל הצורה - scale
328	סיבוב הצורה - rotate
329	אנימציות
332	אנימציות יעילות עם double buffering
332	סיכום



<b>333</b>	<b>פרק 11: נושאים מתקדמים API</b>
333	מה הקשר בין API לבין HTML5? .....
334	מיקום גיאוגרפי - Geolocation .....
334	כיצד פועלת התכונה Geolocation? .....
336	קוד בסיסי ליצירת תוכנית משולבת Geolocation .....
339	אחסון מקומי - Local storage .....
340	לשם מה לשמור את הנתונים בדפדפן? .....
340	דוגמאות למשתמשים בשמירה מקומית .....
340	דוגמה .....
341	הלוגיקה לשמירת מצב .....
342	החלק המרכזי של הקוד .....
343	סיכום .....
344	File API – drag and drop גרירת קבצים מהמחשב האישי אל חלון הדפדפן .....
345	תמיכת דפדפנים .....
345	ההסבר הטכני: File API .....
346	דוגמה לקוד שעושה שימוש ב- File API .....
348	סיכום .....
<b>349</b>	<b>פרק 12: מובייל ורספונסיביות</b>
350	אשליית הרוחב .....
351	אתר נפרד מותאם למובייל .....
355	הבאז סביב המושג Responsive Design (עיצוב מגיב) .....
359	גדלים שונים של מסכים קיימים .....
359	שיקולים טכניים ומגבלות .....
361	היסודות הדרושים לאתר רספונסיבי .....
361	הגדרת Viewport (מעין קנה מידה) .....
361	איך שולטים על ה-viewport? .....
363	הפרמטרים השולטים על ה-Viewport .....
364	פרמטרים נוספים של Viewport .....
364	density - צפיפות פיקסלים במסך .....
366	media queries .....
367	Breakpoints - נקודות עצירה .....
368	performance - שיפור ביצועים .....
369	הדף הרספונסיבי הראשון שלך .....

375	סיכום: הדף הרספונסיבי הראשון שלך
376	שימוש במסגרת קוד - framework
376	Bootstrap
377	היתרונות בשימוש ב- Bootstrap
377	החסרונות בשימוש ב- Bootstrap
377	הטמעת הקוד של Bootstrap באתר האינטרנט שלך
378	פריסת העמוד, הגריד של Bootstrap
381	הטמעת אלמנטים מוכנים מראש של Bootstrap באתר האינטרנט שלך
383	סיכום
385	<b>אינדקס</b>

# הקדמה

ספר זה נכתב על ידי רושו יונית, בעלת ניסיון של כעשור וחצי בהוראה בתכנות מערכות אינטרנטיות ואתרי אינטרנט בטכנולוגיית HTML5 במחלקה להנדסת תוכנה ובמסלול לעיצוב אינטראקטיבי בשנקר, כמו כן במחלקה לניהול מערכות מידע באקדמית של תל אביב יפו ובמכון הטכנולוגי חולון. יונית משמשת כיום ראש התמחות הנדסת Web וענן במחלקה להנדסת תוכנה בשנקר. הצלחת הסטודנטים על פי שיטת לימוד מוגדרת היטב ועקבית, תרמה ליוזמה לריכוז החומר ולכתיבת הספר הזה.

במהלך הלימוד בספר מנסה יונית להקפיד, עד כמה שניתן, על תהליכים מוגדרים וברורים, עבודה יעילה וקוד איכותי, ולהדגיש גם את החשיבות של יעילות הקוד. הרי לא די בכך שהאתר פועל או "עובד" - הוא צריך לשרת באופן מקצועי, יעיל ולהיות בעל משמעות. הספר מצייד אותך בכלים תיאורטיים ומעשיים להבנת רשת האינטרנט, לתכנון ולבנייה של אתרי אינטרנט ומערכות מידע אינטרנטיות בשפה HTML5, משולבת CSS, וביחד עם שפות נלוות כ-JavaScript ו-JQuery לעבודה עצמאית, החל בשלב הגדרת הצרכים ועד להגשת המוצר הסופי. אתה תכיר וגם תבין את הטכנולוגיה במהלך הלימוד של הטקסט והדוגמאות, ולא פחות חשוב - תוך כדי התנסות בפתרון תרגילים.

עם סיום הלימוד במהדורה השלישית של הספר תוכל לבנות אתרי אינטרנט, מערכת מידע מבוססת Web ויישומים רספונסיביים למכשירים ניידים (מובייל) בשפת HTML5, תוך שימוש בטכנולוגיות מתקדמות, לצורך בנייה מקצועית ונקייה, משולבת מדיה, אינטראקטיב וקוד. הספר מתייחס הן למסך הרחב והן למסך הצר של טאבלט ומובייל.

תכני הספר וקטעי הקוד עודכנו לשנת 2018.

HTML5, בשונה מגרסאות קודמות, יותר סלחנית בנושא קידוד, אבל עדיין יש לעבוד בצורה אחידה. עוד על כך תוכל למצוא בספר "Code Complete", מדריך מעשי לפיתוח תוכנה", שיצא בהוצאת הוד-עמי.

## מבנה הספר

לאורך הספר תמצא תרגילים ברמות שונות. לכל התרגילים יש פתרונות. שים לב, לכל תרגיל יש דרכי פתרון שונות. בפתרונות יש גיוון בבחירות הדרכים ואופני הפתרון, כדי לספק לך דוגמאות מגוונות, אולם זכור שיש פתרונות נוספים שיכולים להיות טובים גם הם. בפתרונות יש הקפדה על מינימליזם בקוד, ועל קוד יעיל ואפקטיבי. הפרק העוסק בטפסים משמש כסדנת עבודה, בה בכל שלב של הלמידה מרחיבים את הטופס שבונים ביחד. הספר מחולק לשני חלקים עיקריים: פרקים 1 עד 7 מכסים את כל הנושאים הרלוונטיים להעלאת אתר מקצועי לאוויר. פרקים 8 ואילך עוסקים בנושאים מורכבים שיאפשרו לך לשדרג את

הידע שלך ולהפוך את האתר לרספונסיבי ולאינטראקטיבי בעזרת מדיה, קוד, איתור מיקום הגולש, מתן אפשרות לגרירת קבצים לחלון הדפדפן ועוד.

## קהל היעד

בין אם אתה שואף לדעת לבנות אתר והנך ללא ניסיון או ידע בתחום, ובין אם אתה מנוסה ובקיא בגרסאות קודמות של שפת בניית האתרים HTML, תוכל למצוא בספר זה את התשובות שחיפשת. הספר כולל פרק מבוא המתאר מושגים בסיסיים הכרחיים להבנה, ובמהלך הספר טיפים למי שעבד בגרסאות קודמות. הספר מסיים ברמה מאוד גבוהה של התממשקות ל-API מורכבים, מדיה, תגית הקנבס המפורסמת ובניית אתר רספונסיבי.

## קטעי הקוד

ניתן להוריד את כל קבצי הקוד ב- html ו- css והפתרונות הקיימים מאתר האינטרנט של הוצאת הוד עמי: [www.hod-ami.co.il](http://www.hod-ami.co.il)

בנוסף לקטעי הקוד תמצא בקובץ שייפתח גם קבצי תמונות של מסכים שנמצאים בספר. מכיוון שהספר מודפס ללא צבעים תוכל להיעזר במסך המקורי כדי לראות כיצד הוא נראה עם צבע. מצא את הספר באתר ואת הלינק "קוד מקור" להורדת הקבצים. לחץ עליו ועקוב אחר ההוראות. אם לא תגדיר אחרת, יועתקו הקבצים אוטומטית לדיסק שלך, לתיקייה זו:

**C:\HodAmiBooks\59470\**

תוכל לבחור בעת ההתקנה בכל תיקייה אחרת. לאחר שתוריד את הקבצים מאתר הוד-עמי הם יימצאו (אם לא שניית) בתיקה 59470 שנמצאת תחת HodAmiBooks שנמצאת בכונן C. בספר נתייחס לתיקיית ברירת המחדל הזו. בדרך כלל הפעולות שיש לבצע לאחר הלחיצה על הלינק: לחיצה על הפעל, לחיצה על UnZip, לחיצה על OK ו-Close.

## על יונית רושן

מהנדסת תוכנה, בעלת תואר שני במנהל עסקים ודוקטורנטית בחוג לניהול מידע וידע באוניברסיטת חיפה, בעלת ניסיון מעשי של כמעט 20 שנים בבניית אתרים ומערכות ווביות לחברות הייטק בארץ. יונית ראש התמחות הנדסת Web וענן במחלקה להנדסת תוכנה בשנקר, ומרצה ל-HTML5, לתכנות Web, לעיצוב ממשקים ומסכים, למחשוב ענן ולשירותי מארג האינטרנט במחלקות להנדסת תוכנה, לניהול מערכות מידע ולתקשורת חזותית ב'שנקר' ובאקדמית של תל-אביב יפו. כתבה את מדורי טרנדי באתר 'נענע10' ו-'ניצוצות ברשת' באתר 'הארץ', על מגמות בעיצוב ובטכנולוגיה ברשת. ייסדה, ניהלה ומכרה את 'דיינמיק-ווב', חברה לבניית אתרים ואפליקציות ווב.

אני מאזינה אהבתי את הלימודים והחוקר מהבן מהרה להיז' ארמה מילאניאני מקצוצית.  
יונית

## טבלאות

ראשית, חשוב להדגיש שבכניית אתר או אפליקציה וובית מקצועיים, השימוש בטבלאות צריך להיות מינימלי לצורך הצגת נתונים טבלאיים בלבד, כפי שתקרא בפסקה הבאה. בשום אופן אין להשתמש בתגית הטבלה עבור עימוד ומיקום אלמנטים בעמוד.

טבלאות נועדו להציג מידע בתוך תאים מסודרים, כמו בתכנת MS WORD. דוגמה לנתונים שאפשר לסדר בטבלה: נתוני ביטוח לאומי. בעמודה הימנית יופיעו תעודות הזהות של האנשים. בעמודה השמאלית יופיעו שמות האנשים בהתאמה. זו דוגמה לטבלה פשוטה של שתי עמודות והמון שורות. לשם כך הטבלאות נועדו. לטבלאות תכונות מיוחדות שמאפשרות לפרוס תאים (merge cells) ולכן, עם השנים, החלו להשתמש בטבלאות לצרכים נוספים, כגון ליצור layout מבנה העמוד ומיקום האלמנטים בו. בשיטה זו יש פגיעה חמורה מאוד במינימליזם של הקוד, בהפרדת התוכן מהתצוגה (מאחר ותגיות *התוכן* של הטבלה הפכו להיות תגיות שמשפיעות על *התצוגה*) ובחוסר המשמעות של הקוד. שיטה זו הייתה בשימוש שנים רבות ויצאה מכלל שימוש הרבה לפני ש-HTML5 "נולדה". לכן, יש לזכור להשתמש בטבלאות בתבונה ובמידה, כשצריך בלבד. זו שיטה ישנה שטובה רק למקרים בסיסיים של הצגת מידע טבלאי.

כדי להציג מידע טבלאי, תוכל ליצור שורות, ובשורות ליצור תאים, בהם תציב טקסטים ותמונות.

תגית הטבלה היא `<table></table>`

כל שורה מוגדרת על ידי תגית השורה `<tr>` ומסתיימת בהופעת תגית הסגירה של השורה: `</tr>`.

בין תגית הפתיחה לבין תגית הסגירה של השורה מופיעים כל התאים באותה השורה. התגית המגדירה תא היא `<td>` `table data` בו תוכל להציג את כל סוגי המידע שלמדת עד היום להציג: קישורים, טקסטים, תמונות ואפילו טבלאות פנימיות.

לדוגמה, נבחן את הטבלה הראשונה:

```
<table border="1">
  <tr>
    <td> row 1, cell 1</td>
    <td> row 1, cell 2</td>
  </tr>
  <tr>
    <td> row 2, cell 1</td>
    <td> row 2, cell 2</td>
  </tr>
</table>
```

להלן התוצאה בדפדפן:

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

## תאים ריקים בטבלה

טבלה עם תאים ריקים אינה מוצגת כראוי.

```
<table>
<tr>
  <td>row 1, cell 1</td>
  <td>row 1, cell 2</td>
</tr>
<tr>
  <td>row 2, cell 1</td>
  <td></td>
</tr>
</table>
```

ולהלן התוצאה בדפדפן:

row 1, cell 1	row 1, cell 2
row 2, cell 1	

תוכל לראות שהתא הריק מוצג באופן בלתי תקני. לכן, יש להוסיף `&nbsp;` (רווח) בתאים שמיועדים להישאר ריקים:

```
<table border="1">
<tr>
  <td>row 1, cell 1</td>
  <td>row 1, cell 2</td>
</tr>
<tr>
  <td>row 2, cell 1</td>
  <td>&nbsp;</td>
</tr>
</table>
```

ולהלן התוצאה בדפדפן:

row 1, cell 1	row 1, cell 2
row 2, cell 1	

## מרווחים בין תאים ובתוך התאים

ניתן להגדיר רווח בין תא לתא (הגבול בין התאים) וניתן להגדיר שוליים פנימיים בתוך התא.

לקביעת הריווח בין התאים הוסף את המאפיין **cellspacing** לתגית הפתיחה <table> דוגמה לריווח בין התאים:

```
<table cellspacing="4">
```

בדוגמה לעיל אתה מגדיר טבלה שיש רווח של 4 פיקסלים בין התאים לרוחב ולגובה (זהו בעצם רוחב המסגרת שמפרידה בין התאים).

לקביעת השוליים הפנימיים בכל תא, הוסף את המאפיין **cellpadding**.

דוגמה לריווח בתוך התאים:

```
<table cellpadding="4">
```

בדוגמה לעיל אתה מגדיר טבלה שיש רווח של 4 פיקסלים בין שוליי התא לבין תוכנו.

## פריסת תאים Merge Cells

על מנת לפרוס תא אחד על פני שני תאים לרוחב, או על פני שתי שורות לגובה, יש להשתמש במאפיינים **rowspan** ו-**colspan** בתגיות הפתיחה של התאים שרוצים למזג.

**colspan**: מגדיר את מספר התאים (עמודות) שעליהן יתפרס התא לרוחב.

לדוגמה, כדי לפרוס תא על פני שלושה תאים (עמודות) לרוחב, נגדיר: <td colspan="3">

**rowspan**: מגדיר את מספר השורות שעליהן יתפרס התא לגובה.

לדוגמה, כדי לפרוס תא על פני שתי שורות, נגדיר: <td rowspan="2">

קוד exercise\_2.15.html מהווה דוגמה לטבלה, בה התא בשורה הראשונה פרוס על פני כל העמודות לרוחב, והתא הראשון משמאל בשורה השנייה מתפרס על פני שתי שורות לגובה.

```

<!DOCTYPE html>
<html>
  <head>
    <title>Tables</title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
  </head>
  <body>
    <table border="1" cellpadding="3" cellspacing="3"
style="width:400px">
      <tr>
        <td colspan="3"> התא הראשון מתפרס על פני שלושה תאים
לרוחב</td>
      </tr>
      <tr>
        <td rowspan="2"> התא הראשון בעמודה זו מתפרס על פני שתי
שורות</td>
        <td> התא השני בשורה</td>
        <td><a href="contact.html">קישור לעמוד צור קשר</td>
      </tr>
      <tr>
        <td> זה התא השני בשורה השלישית. בקוד הוא כתוב כתא
</td>
        <td> הראשון, כי התא הראשון "תפוס" על ידי התא מהשורה שמעל
</td>
        <td> תא שלישי בשורה. בקוד הוא כתוב כתא השני</td>
      </tr>
    </table>
  </body>
</html>

```

תרשים 2.24 מציג את התוצאה בדפדפן אינטרנט אקספלורר. הטבלה תיראה אותו הדבר גם על הדפדפנים האחרים:

השורה הראשונה מתפרסת על פני שלושה תאים		
התא הראשון בעמודה זו מתפרס על פני שתי שורות	התא השני בשורה	<a href="#">קישור לעמוד צור קשר</a>
זה התא השני בשורה השלישית. בקוד הוא כתוב כתא הראשון, כי התא הראשון "תפוס" על ידי התא מהשורה שמעל	תא שלישי בשורה. בקוד הוא כתוב כתא השני	

תרשים 2.24



## סיכום

אם הגעת לכאן, משמע שעמדת בהצלחה בלימוד יסודות HTML כשפה הראשית בעולם לבניית אתרים, והכרת את הכללים והחוקים של HTML5 בפרט.

אתה יודע כעת לבנות דף HTML5 תקני, לכתוב בו טקסטים בגדלים שונים, להוסיף תמונות מסוגים שונים, להגדיר קישורים פנימיים וחיצוניים, ליצור רשימות ועוד. הדרך עוד ארוכה, אך כשהיסודות חזקים אפשר להתקדם בבטחה.

בפרק הבא נסקור את התגיות החדשות של HTML5, בזכותן תוכל להגדיר אזורים שונים בדף האינטרנט ולהתחיל לראות את המבניות של דף, של ממשק או של משחק.



# 3

## תגיות המבנה החדשות

---

פרק זה הינו אחד החשובים ללומד HTML5, הן למתחיל והן למי שכבר התנסה בכתיבת דפי HTML בגרסאות קודמות. כאשר מפתחי השפה קבעו את ההגדרות שלה, הם לקחו בחשבון צרכים עכשוויים וחידושים טכנולוגיים עתידיים, ככל שיכלו להעריך אותם. אחד הצרכים שנלקח בחשבון הוא השימוש בתגיות בעלות משמעות ליצירת מבניות בדף הקוד. אם אתה בעל ניסיון בגרסאות קודמות, בוודאי תסכים שמספר תגיות `<div>` שהשתמשת בהן היה גדול וחסר כל משמעות. על כן, בפרק זה מרוכזות התגיות החדשות העיקריות ב-HTML5 שמטרתן לסייע ביצירת מבנה הדף, והוא כולל גם נושאים חשובים אחרים שיש לתת עליהם את הדעת בעת כתיבת דפי HTML5.

בפרק זה נבחן את הנושאים הבאים:

- התגית `<header>`
- התגית `<footer>`
- התגית `<section>`
- התגית `<main>`
- התגית `<article>`
- התגית `<nav>`
- התגית `<aside>`
- אזורים יעודיים
- תגיות נוספות
- מודל התוכן של HTML5

### **מנוסה בגרסאות קודמות של HTML?**

כל התגיות בפרק זה חדשות עבורך. בחלק האחרון של הפרק תמצא פירוט של מודל התוכן החדש של HTML5, אשר מחליף את הגדרות block ו-inline שאתה מכיר.

# תגיות חדשות ב-HTML5 ליצירת מבניות בעלת משמעות

בפרק 6 תלמד על כך שדרושות תגיות מעטות ונדון במשמעות שלהן, ותלמד על החשיבות שהקנו לכך יוצרי ומפתחי השפה HTML5. בפרק זה נבחן את התגיות החדשות של השפה.

שים לב, אלו הן תגיות מבנה. הן יוצרות וקובעות את המבנה של הדף ואת המרכיבים העיקריים שלו. לפנינו אם כן, המקבילה לשלד של בניין.

## <header> התגית

התגית <header> (כותרת עליונה) מגדירה בדרך כלל את האזור העליון של הדף, לעתים היא גם כוללת תפריט עליון אופקי, באנר, לוגו וכד'. התגית יכולה להכיל כמעט כל תגית אחרת למעט <header> ו-<footer>. מנועי החיפוש מודעים לכך, שתוכן שנמצא בתוך תגית זו חשוב, וזו סיבה מספיק חשובה שגם המשתמש, יוצר דף הקוד, יראה זאת כך. ניתן להגדיר יותר מתגית <header> אחת בדף.

דוגמה לאופן השימוש בתגית תוכל לראות בקוד `exercise_3.1.html`:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Yonit</title>
  </head>
  <body>
    <header>
      <h1>Profile Company</h1>
      <h2>Goals</h2>
    </header>
  </body>
</html>
```

## <footer> התגית

התגית <footer> (כותרת תחתונה) מגדירה את החלק התחתון של הדף. באתרי אינטרנט, כלולים בה בדרך כלל קישורים שונים כמו לדוגמה, פרטי יצירת קשר וזכויות יוצרים.

```
<footer>footer links</footer>
```

## <section> התגית

התגית <section> מגדירה מקטע, אזור או חלק בדף הקוד, כגון פרק או ידיעה חדשה בתוך חלון חדשות. התגית אינה מוסיפה לעיצוב, אלא רק למבניות ולמשמעות הקוד. היא יכולה

לאגד בתוכה מספר תגיות תוכן, ובכלל זה את <article>. קיים דמיון בינה לבין התגית <article> שמוצגת להלן. ההבדלים המרכזיים הם, שהתגית <section> יכולה להכיל את התגית <main> המוצגת בהמשך, ואילו התגית <article> אינה יכולה להכילה. כמו כן, התוכן שנמצא בתוך התגית <section> לא חייב להיות בעל משמעות עצמאית, אם מפרידים אותו מהעמוד לחלוטין.

לדוגמה - קוד exercise\_3.2.html:

```
<body>
  <section>
    <a href="help.html">help</a>
    <h1>TOKYO</h1>
    <h2>CITY OF NEON LIGHTS</h2>
    <article>
      <p>
        Tokyo's shining neon lights mark a strong visual impact
      </p>
    </article>
    <section>
      
    </section>
  </section>
</body>
```

כפי שתוכל לראות, התגית <section> מכילה מספר תגיות, כולל קישור, תמונה, כותרות ופסקה, מתוך צורך לוגי של הגדרת אזור בעמוד. אם תיקח את התגית <section> על תכולתה למסמך אחר, נניח למסמך וורד, לא דווקא תהיה משמעות זהה לתוכן.

בהמשך הפרק ראה דוגמה בתרשים 3.2 לחלוקת עמוד לאזורים ייעודיים, וביניהם <section>.

## התגית <article>

התגית <article> מגדירה אזור או חלק בדף הקוד שיש לו שימוש עצמאי נוסף, או מפרסמים אותו, כמו פוסט בפורום, מגזין או ניוזלטר, מאמר, בלוג, תגובה, הערה (comment) או כל תוכן עצמאי אחר. כלל האצבע לשימוש בתגית זו, הוא היכולת שלה להיות עצמאית בעמוד אחר. משמע, אם אתה יכול לקחת את התוכן שבתגית, ולמקם אותו בעמוד אחר, ועדיין תהיה לתוכן משמעות, אזי התגית <article> מצוינת. כמו <section>, גם תגית זו אינה מוסיפה לעיצוב, אלא רק למבניות ולמשמעות הקוד. התגית יכולה לאגד בתוכה מספר תגיות תוכן, כולל <section>, אך אינה יכולה להכיל את התגית <main>. אם המטרה היא רק ליצור מלבן עם תמונת רקע לדוגמה, וללא תוכן שיש לאגד, אפשר להשתמש בתגית <div> הישנה, שנסקור אותה בהמשך.

## הגדרות העיצוב הבאות רשומות בקובץ CSS החיצוני:

### הגדרת עיצוב 1

```
body {  
  background-color:orange;  
}
```

הגדרת עיצוב 1 קובעת צבע רקע כתום לכל גוף המסמך, מכיוון שההגדרה נקבעה עבור התגית `<body>` שבמסמך HTML.

### הגדרות עיצוב 2

```
header {  
  background-color:#999999;  
  width:150px;  
}
```

השורה הראשונה בהגדרות עיצוב 2 קובעת שיהיה צבע רקע אפור, שערכו 999999 (כפי שנדגם בתוכנה גרפית), עבור כל התגיות header שקיימות במסמך HTML. השורה השנייה בהגדרות עיצוב 2 קובעת את רוחב האלמנט. מיהו האלמנט שנקבע רוחבו? התגית header שבמסמך HTML. רוחב התגית יהיה 150 פיקסלים. מלל שיהיה כתוב בתוך התגית הזו, יופיע בתוך אזור צר זה.

הנה דוגמה נוספת (הספרור הוא רק לצורך התייחסות ואינו חלק מהקוד):

### הגדרות עיצוב 3

```
h1 {  
  1 background-color:white;  
  2 font-family:Arial;  
  3 font-size:18px;  
  4 color:#1690ee;  
  5 padding-top:17px;  
  6 text-align:center;  
  7 width:180px;  
}
```

הגדרות עיצוב 3 חלות על כל תגיות h1 שכתובות במסגרת התגית body.

בשורה 1 צבע הרקע מוגדר לבן. אתה יכול לשאול מדוע להגדיר צבע רקע לבן, אם צבע הרקע של המסמך הוא תמיד לבן כברירת המחדל. שאלתך נכונה. אך התגית h1 תלויה בהגדרות שהוגדרו בתגית האב שלה, שהינה התגית body. כפי שראינו, לתגית body נקבע צבע רקע כתום, ולכן אם ל-h1 מוגדר רקע לבן, יוצג מלבן בעל רקע לבן על הרקע הכתום של המסמך.

בשורה 2 מוגדר סוג גופן אריאל.

**בשורה 3** מוגדר גודל גופן 18 פיקסלים.

**בשורה 4** מוגדר צבע גופן (אם כתוב color בלבד, הכוונה תמיד לצבע הגופן).

**בשורה 5** מוגדר רווח עליון של 17 פיקסלים. הסבר נוסף על רווחים ושוליים פנימיים וחיצוניים תוכל לקרוא בפרק 5. תוכל לראות שרווח זה "דוחף" כלפי מטה את הטקסט של הכותרת.

**בשורה 6** מוגדר מרכז טקסט הכותרת.

**בשורה 7** נקבע רוחב התגית ל-180 פיקסלים. שים לב, שללא הגדרת הרוחב, יוצג צבע הרקע לכל הרוחב. הסיבה לכך היא, שתגית הכותרת מוצגת על פני כל רוחב המסך כברירת המחדל שלה, כפי שלמדת בפרק 3.

#### הגדרות עיצוב 4

```
h2 {  
  1. font-size:14px;  
  2. font-family:Verdana;  
  3. color:#1690ee;  
  4. background-image:url(../images/bgr2.jpg);  
  5. padding-left:10px;  
}
```

פירוט הגדרות עיצוב 4:

**בשורות 1, 2, 3** מוגדר לכל התגיות במסמך מסוג h2 גופן בגודל 14 פיקסלים, סוג גופן ורדנה וצבע (של הגופן).

**בשורה 4** תוכל לראות שבמקום להגדיר צבע רקע (background-color), הגדרנו תמונת רקע (background-image). הגדרת תמונת רקע ארוכה ומסורבלת, אך מכיוון שהשימוש בה נפוץ, תלמד אותה במשך הזמן. הסבר נוסף תוכל למצוא בהגדרות לפי נושאים בהמשך הפרק.

מכיוון שלא הוגדר אחרת, התמונה תשכפל עצמה לאורך ולרוחב כל אזור התגית.

להבדיל מהגדרות העיצוב של h1, לא מוגדר פה רוחב, ולכן תוכל לראות שצבע הרקע של כותרת המשנה פרוס על פני כל רוחב מסך המחשב.

**בשורה 5** מגדירים רווח משמאל לטקסט. ללא רווח זה, יהיה הטקסט שבכותרת המשנה צמוד לשפה השמאלית של תגית האב, המקטע (section), וזה לא נראה יפה.

שים לב שבמסמך HTML5 המפורט בקוד exercise\_4.6.html יש שתי תגיות <h2>, ושתייהן מקבלות את אותן הגדרות עיצוב.

## הגדרות עיצוב 5

```
nav {
  background-color: rgba(0,0,0,0.8);
  width:100px;
}
```

בהגדרה 5 נקבע צבע רקע שחור בשקיפות של 80% (בשיטת RGB שהוזכרה בפרק 2 ובסעיף קודם בפרק זה) וברוחב 100 פיקסלים.

## הגדרות עיצוב 6

```
ul {
  list-style-type:none;
}
```

הגדרה 6 זו מיועדת בעיקר להגדרת רשימות ללא סימן התבליט.

## הגדרות עיצוב 7

```
a {
  1. text-decoration: none;
  2. color: yellow;
}
```

הגדרות עיצוב 7 חשובות וגם שימושיות מאוד לעיצוב קישורים. מהגדרות אלו יושפעו כל תגיות a שבמסמך, אשר כזכור הן תגיות קישור. ראית בתרגילים קודמים שברירת המחדל של תגית הקישור היא טקסט בצבע כחול או סגול עם קו תחתי. בהגדרות עיצוב 7 בשורה 1 תוכל לראות כיצד אפשר להוריד את הקו התחתי על ידי כתיבת "none". בשורה 2 קובעים את צבע הגופן לצהוב.

שים לב שבמסמך HTML5 המפורט בקוד exercise\_4.6.html יש ארבע תגיות a, וכולן מקבלות את אותן הגדרות עיצוב.

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS</title>
    <link rel="stylesheet" href="includes/style.css">
  </head>
```



```

<body>
  <header>
    HTML5 Tech News
  </header>
  <section>
    <h1>Top News</h1>
    <h2>Check out the latest news</h2>
  </section>
  <nav>
    <ul>
      <li><a href="index.html">Home Page</a></li>
      <li><a href="about.html">Vision</a></li>
      <li><a href="vision.html">Contact</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </nav>
  <h2>Same style as the above h2 tag</h2>
</body>
</html>

```

בתרשים 4.5 תוכל לראות את התצוגה בדפדפן עבור הגדרות העיצוב 1 עד 7 שראית קודם.



תרשים 4.5

# הגדרות CSS שימושיות לפי נושאים

המאפיינים העיקריים נחלקים לסיווגים עיקריים אחדים:

## רקעים

### background-color - צבע רקע

תיאור: הגדרת צבע רקע לתגית.

הגדרת ברירת מחדל: שקוף.

אפשרויות:

1. `background-color:white;`
2. `background-color:rgba(0,0,0,0.8);`
3. `background-color:#ffffff;`

שורה 1 מגדירה צבע רקע לבן אחיד בעזרת המילה השומרה `white`.

שורה 2 מגדירה צבע רקע לבן ב- 80% נראות. משמע, שקיפות של 20%.

שורה 3 מגדירה צבע רקע לבן בעזרת ערך הקסדצימלי `#ffffff`.

מאחר ובגרסה 3 של CSS (CSS3) ניתן להגדיר שקיפות בעזרת פקודת `opacity` (על כך בפרק 8), אין כמעט שימוש בהגדרה שבשורה 2.

### background-image - תמונת רקע

תיאור: הגדרת תמונת רקע עבור התגית.

הערה: ניתן להגדיר יותר מתמונת רקע אחת לתגית. הסבר כיצד לעשות זאת תמצא בפרק 8.

הגדרת ברירת מחדל: אין.

אפשרויות:

```
background-image:url(../images/bgr2.jpg);
```

בהגדרה זו יש לשים לב לנתיב של התמונה שרשום בין הסוגריים. מקובל להגדיר נתיבים לתמונות מנקודת הייחוס של מסמך HTML. לכן, הנתיב אל התמונות למשל, הוגדר בדרך זו: ``, כשהוא מכוון אל התיקייה `images`.

שים לב לשינוי על ידי שימוש ב-CSS: את תגית התמונה עליך להגדיר בקובץ HTML, ואת תמונת הרקע עליך להגדיר בקובץ CSS ששמור בתיקייה `includes`. לכן, באופן יחסי לקובץ CSS, תוכל לראות שקובץ התמונה נמצא ברמה אחת גבוהה יותר בתיקיות המחשב (רמה

אחת למעלה בתיקיות של המחשב היא התיקייה שבה שמור קובץ HTML שלך), וממנה אל התיקייה images שהגדרנו עוד בפרק 3.

### **background-position - מיקום תמונת רקע**

תיאור: הגדרת המיקום של תמונת הרקע של התגית. מגדירים את המיקום על פי ציר X ועל פי ציר Y.

הגדרת ברירת מחדל: נקודה שמאלית עליונה של המסך.

להלן מספר דוגמאות לשימוש במילים שמורות, בערכי פיקסלים ובאחוזים.

תמונת הרקע צמודה לצד שמאל (ציר X) ולמעלה (על ציר Y):

```
background-position:left top;
```

תמונת הרקע צמודה לצד ימין (ציר X) ולאמצע (על ציר Y):

```
background-position:right center;
```

מיקום תמונת הרקע הוא 10 פיקסלים משמאל (על ציר X) ו-200 פיקסלים מלמעלה (על ציר Y):

```
background-position:10px 200px;
```

מיקום תמונת הרקע הוא 50% מצד שמאל (ציר X) ו-50% מלמעלה (על ציר Y):

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

### **background-repeat - שכפול תמונת רקע**

תיאור: הגדרת תכונת השכפול של תמונת הרקע עבור התגית.

הגדרת ברירת מחדל: שכפול לרוחב ולגובה של התגית.

יש כמה אפשרויות כפי שמפורט להלן.

שכפול לרוחב ולגובה: `background-repeat:repeat;`

שכפול לרוחב בלבד: `background-repeat:repeat-x;`

שכפול לגובה בלבד: `background-repeat:repeat-y;`

ללא שכפול, ואז תוצג פעם אחת בלבד: `background-repeat:no-repeat;`

### **מסגרות**

מקובל לבטל לחלוטין את המסגרות, אך לעתים תרצה להגדיר למסגרת צבע מסוים, עובי וסגנון. האפשרויות נחלקות לשלוש קבוצות.

## border-width - עובי מסגרת

הגדרת עובי הקו של המסגרת. לכל אחד מקווי המסגרת ניתן להתייחס בנפרד, ולהגדיר עובי במילים שמורות (thin, medium, thick; ברירת המחדל היא medium), או על ידי ערכי פיקסלים.

```
border-top-width:medium;
border-right-width:thick;
border-bottom-width:thin;
border-left-width:5px;
```

ניתן לקבוע רוחב אחיד לכל קווי המסגרת:

```
border-width:8px;
```

לחילופין, ניתן לקבוע לכל קו מסגרת עובי שונה, ולכתוב את הערכים בשורה אחת, משמאל לימין. הערכים הם עבור קו המסגרת העליון, הימני, התחתון והשמאלי, עם כיוון השעון:

```
border-width:8px 5px 10px 0px;
```

## border-style - סגנון מסגרת

הגדרת סגנון הקו של המסגרת.

לכל אחד מקווי המסגרת ניתן להתייחס בנפרד, ולהגדיר סגנון על ידי מילים שמורות.

```
border-top-style: dotted;
border-right-style: dashed;
border-bottom-style: groove;
border-left-style: solid;
```

לחילופין, ניתן לקבוע לכל צד הגדרה שונה, ולעשות זאת בשורה אחת:

```
border-style: dotted dashed groove solid;
```

הכתיבה משמאל לימין מתייחסת לקו המסגרת העליון, הימני, התחתון והשמאלי, עם כיוון השעון.

האפשרויות הקיימות לסגנון קו המסגרת:

none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset

אין הגדרה של ברירת מחדל.

## border-color - צבע מסגרת

הגדרת הצבע של קווי המסגרת.

לכל קו מסגרת ניתן להתייחס בנפרד, ולהקבוע עבורו את הצבע על ידי מילים שמורות לצבעים או על ידי מספר הקסדצימלי, או לציין שקיפות (transparency) כפי שמוסבר בסעיף הגדרות צבע. אין הגדרה של ברירת מחדל.

```
border-top-color: red;  
border-right-color: blue;  
border-bottom-color: #000000;  
border-left-color: #999999;
```

ניתן לקבוע לכל אחד מקווי המסגרת את אותו ערך צבע:

```
border-color: green;
```

לחילופין, ניתן לקבוע לכל קו ערך צבע שונה, ולכתוב זאת בשורה אחת. הכתיבה משמאל לימין מתייחסת לקו המסגרת העליון, הימני, התחתון והשמאלי, עם כיוון השעון:

```
border-color: red blue #00000 #999999;
```

הגדרת עיצוב למסגרת עבור צבע, עובי וסגנון שכתובים בשורה אחת:

```
border: green solid 2px;
```

הגדרה זו תיצור מסגרת ירוקה בעובי 2 פיקסלים בסגנון קו מלא (solid) בכל הצדדים.

## טיפוגרפיה: הגדרות עיצוב עבור טקסטים וגופנים

טיפוגרפיה היא אומנות עיצוב הטקסטים. הרווחים בין האותיות, השורות, סוג הגופן וכו'. בכל אתר אינטרנט ואפליקציה יש תוכן הכולל טקסטים רבים. בבואנו לעצב את הטקסטים אנו מפרידים בין טקסט 'חי' לבין טקסט 'תמונה'.

### שימוש בטקסט 'חי'

טקסט 'חי' הוא טקסט שכתוב בין תגיות HTML ואינו חלק מתמונה. היתרונות של טקסט 'חי' רבים: מנועי החיפוש קוראים אותו, ניתן לשנות אותו בקלות, להעתיק ולהדביק אותו, לשנות את גודל הגופן, ניתן לעדכן במערכות לניהול תוכן אוטומטיות, ועוד.

היכן יש טקסט חי? כמעט בכל מקום ובכל אתר: בקישורים שבתפריטים, בתוכן, בכותרות, בבאנרים טקסטואליים, בחדשות, ועוד.

אם כן, היכן יש שימוש בתמונות? לוגו, תמונות רקע המהוות את עיצוב האתר, ותמונות הכלולות בתוכן.

## Web-safe fonts - גופן מערכת

עד לאחרונה נהגו להשתמש בגופני מערכת לעיצוב אתרים, ובעיקר השתמשו בגופן Arial. נהגו כך, בין השאר, מכיוון שבמערכות הפעלה שונות מותקנים גופנים שונים והיה צורך להשתמש בגופנים שמקובלים ברוב המערכות, אם לא בכולן. יש לזכור, שהדפדפן מחפש את הגופן במחשב הלקוח, כדי להציגו.

מקובל לכלול בקוד CSS רשימת גופנים, ואם הדפדפן אינו מוצא את הגופן הרצוי, הוא מחפש את הבא ברשימה כדי להתקין אותו, עד שהוא מגיע לסופה. אם הדפדפן לא ימצא אף אחד מסוגי הגופנים המוגדרים לו, הוא יציג את גופן ברירת המחדל שלו, סביר להניח שזה יהיה Times New Roman. דוגמה להגדרה:

```
font-family:Arial, Verdana, David, Tahoma;
```

## font-size, font-family, color - צבע הגופן, משפחה וגודל

קוד exercise\_4.7.html כולל דוגמת הגדרה של שלושה ערכים:

1. color - צבע הגופן

2. font-family - סוג הגופן

3. font-size - גודל הגופן

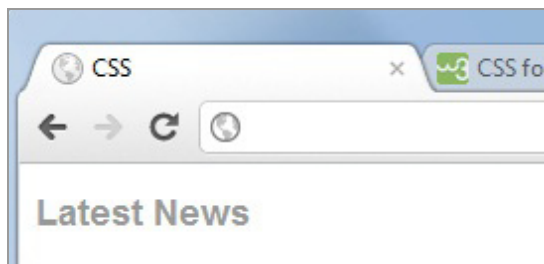
הגדרות העיצוב של הגופן נראות כך במסמך CSS:

```
h1
{
  1. color:#999999;
  2. font-family:Arial, Verdana, Tahoma;
  3. font-size:18px;
}
```

התגיות נראות כך במסמך HTML5:

```
<section>
<h1>Latest News</h1>
</section>
```

בתרשים 4.6 תוכל לראות את התוצאה בדפדפן:



תרשים 4.6

## מיקום יחסי - Relative positioning

כאשר ממקמים אלמנט באופן יחסי ומזיזים אותו ממקומו, ניתן יהיה להזיזו באופן יחסי מנקודת ההתחלה שלו ולהחזירו אליה, כי המקום שלו נשמר. נשתמש בדוגמה: עשרה ילדים עומדים על במה זה לצד זה בטקס כלשהו. כל ילד בתורו צריך להתקדם שני צעדים לפנים ולקרוא את תפקידו. המקום מאחור נשמר עבורו, ושאר הילדים אינם זזים ממקומם. לכן, כשהילד שצעד קדימה מסיים לקרוא, הוא צועד שני צעדים לאחור אל המקום שנשמר לו.

מיקום יחסי של תגיות מתנהג באופן דומה. אם אתה מזיז תגית ממקומה המקורי, כמו שהילד זז, מקומה המקורי יישמר.

בדוגמה הבאה, נגדיר תזוזה של 15 פיקסלים משמאל ומלמעלה. שים לב, שבדוגמאות של מיקומים אני משתמשת בתגיות `<div>`, אשר ברירת מחדל שלהן היא `block`.

בדוגמה זו, ניצור מסמך HTML5 ובו שלוש תגיות `<div>`. לכל תגית נגדיר מאפיין `id` עם ערך `box1`, `box2`, `box3` בהתאמה.

להלן הגדרות העיצוב של `#box1`:

```
#box1 {
  border: 1px solid black;
  margin: 0px;
  padding: 0px;
  height: 15px;
  background: grey;
}
```

הגדרות העיצוב של `#box3` זהות למעט צבע רקע כסף.

קוד `exercise_5.4.html` מציג את הקוד במלואו. התגית השנייה (`#box2`), בעלת אותן הגדרות עיצוב כמו `#box1` ו-`#box3` למעט המיקום שלה. היא הוזזה באופן יחסי 20 פיקסלים מלמעלה ו-20 פיקסלים משמאל באופן יחסי למיקום המקורי שלה בזרימה של המסמך:

```
#box2 {
  position: relative;
  top: 20px;
  left: 20px;
  border: 1px solid black;
  margin: 0px;
  padding: 0px;
  height: 15px;
  background: black;
  color: white;
}
```

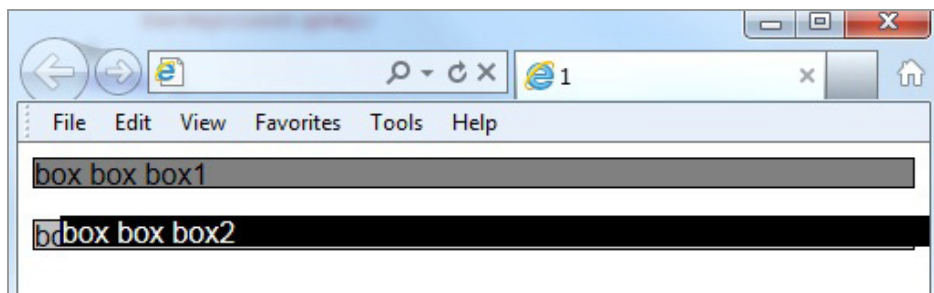
```

<body>
  <div id="box1">box box box1</div>
  <div id="box2">box box box2</div>
  <div id="box3">box box box3</div>
</body>

```

לולא הזזת #box2, באופן יחסי למיקומו המקורי, היו מוצגים בדפדפן שלוש תגיות ה- <div> זו תחת זו באופן רגיל.

התוצאה, כפי שתוכל לראות בתרשים 5.11, מוכיחה שהאלמנט המקורי box2, ממשיך להחזיק בשטח המקורי שלו כחלק מהזרימה הכללית וממשיך להשפיע עליה, ויכול לחפוף אלמנטים אחרים.

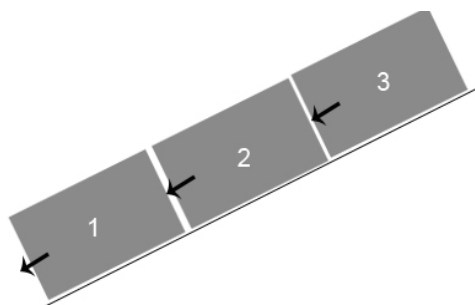


תרשים 5.11

## מיקום מוחלט - Absolute positioning

כפי שראית, מיקום יחסי נחשב חלק מהזרימה הרגילה (normal flow), מכיוון שהאלמנט מוצב באופן יחסי למיקומו המקורי. בניגוד לו, מיקום מוחלט מעביר את האלמנט מחוץ לזרימה הרגילה, ולכן לא נשמר עבורו המקום בזרימה הכללית, והוא אינו משפיע עליה. שאר האלמנטים מתנהגים כאילו האלמנט הזה בכלל אינו קיים.

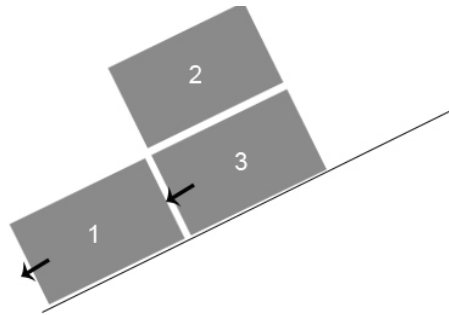
תוכל לדמיין מסוע משופע במפעל לייצור קופסאות. כל הקופסאות נמשכות כלפי מטה זו אחר זו, כמתואר בתרשים 5.12.



תרשים 5.12



נניח שמגיע סבל ומרים את קופסה 2. מה יקרה לקופסה 3? ברור שהיא תימשך למטה ותתפוס את המקום של קופסה 2 שהוצאה מהמסוע, כפי שתוכל לראות בתרשים 5.13.



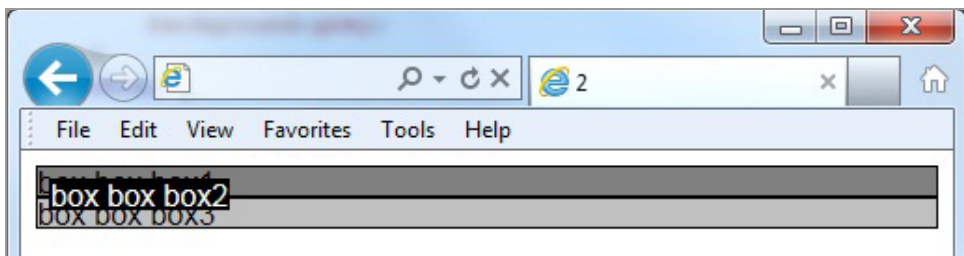
### תרשים 5.13

מה שקרה במסוע המשופע במפעל יכול להמחיש את מה שקורה לתגיות. התגיות תלויות זו בזו וממשיכות זו את זו ברצף הזרימה של האלמנטים במסמך. כאשר אתה מגדיר מיקום מוחלט לתגית מסוימת, כאילו הורית להוציא אותה מהמסוע. שאר התגיות יתעלמו כעת מקיומה ויתפסו את החלל שהתפנה.

בקוד `exercise_5.5.html` שונתה רק הגדרת המיקום של התגית השנייה ונהפוך אותה מיחסית למוחלטת. הנה כך:

```
#box2 {  
  position: absolute; /*changed*/  
  top:15px;  
  left:15px;  
  border: 1px solid black;  
  margin:0px;  
  padding:0px;  
  height:15px;  
  background:blue;  
}
```

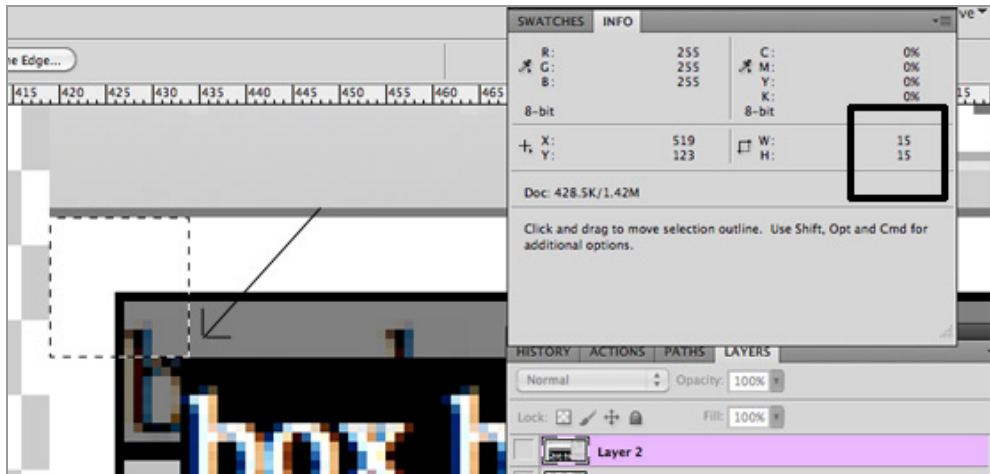
התוצאה מוצגת בתרשים 5.14. האלמנט השני (`box2`) הוזז 15 פיקסלים מלמעלה ו-15 פיקסלים משמאל.



### תרשים 5.14

הערה: שים לב לשינוי ברוחב. הוא מפסיק להיות 100% באופן אוטומטי, ומתאים את עצמו לרוחב הטקסט. עם זאת, ניתן להגדיר לו רוחב, להבדיל מאלמנט שיש לו הגדרת inline כברירת מחדל, שלא ניתן להגדיר לו רוחב.

בתרשים 5.15 תוכל לראות מהיכן נמדדים 15 הפיקסלים. זהו צילום מסך של תוכנת פוטושופ של Adobe. בריבוע מקווקו בצד שמאל למעלה סומנו 15 פיקסלים מהפינה השמאלית העליונה, שהיא פינת ברירת המחדל של הדפדפן, ועד הפינה השמאלית העליונה של box2 שהוזה.



תרשים 5.15

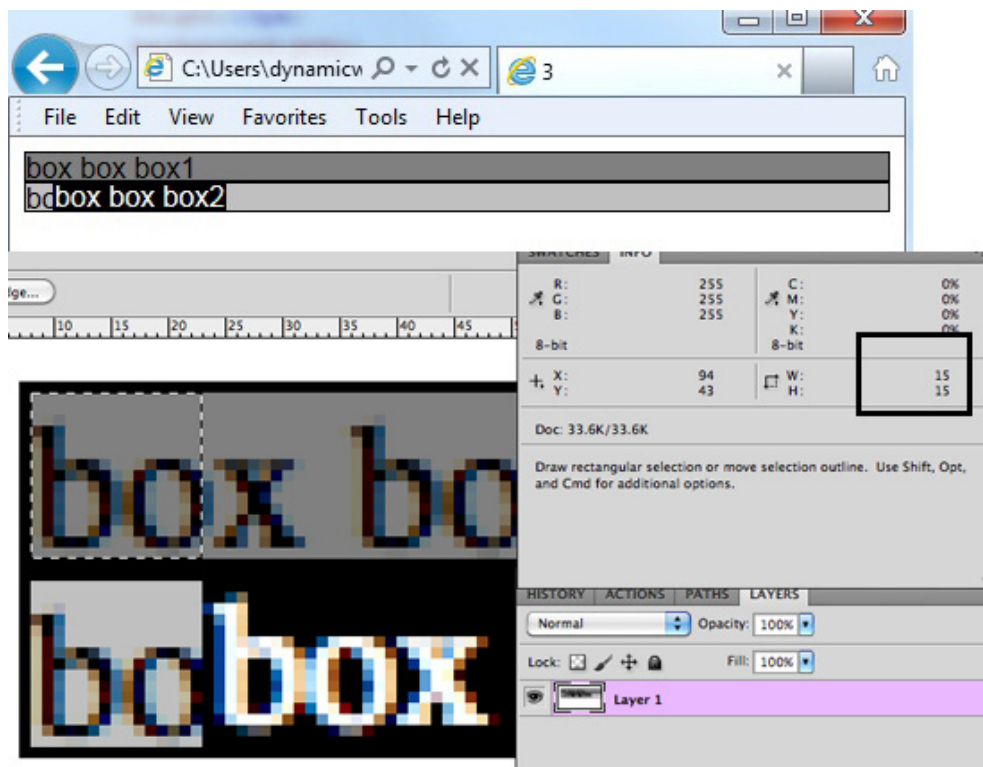
נוכל לסכם בדרך זו את נושא הזרימה ומיקום האלמנטים בדף: אלמנט שממוקם באופן מוחלט, נמצא באופן "יחסי" לאלמנט האב הקרוב ביותר שהוגדר לו מיקום ואם אין כזה, הוא יהיה באופן יחסי לאלמנט הראשון בדף. אלמנט שממוקם באופן יחסי נמצא למעשה באופן "יחסי" למקום המקורי שלו בזרימה הרגילה של המסמך. כדי להבין את ההשפעה של הגדרות תגית האב על תגית הילד בעניין מיקומים יחסיים ומוחלטים, הכנס את div2 אל תוך div1. הגדר מיקום יחסי עבור div1 ומיקום מוחלט עבור div2. הגדר צבע רקע שונה לכל אחד מהם כדי שתוכל לראות את המיקומים טוב יותר.

קוד exercise\_5.6.html מציג את השינוי שנדרש. תחילה מוצגות הגדרות CSS ולאחריהן מוצג קוד HTML:

```
#box1 {
  position: relative; /*changed*/
  border: 1px solid black;
  margin: 0px;
  padding: 0px;
  height: 15px;
  background: grey;
}
#box2 {
  position: absolute;
  top: 15px;
  left: 15px;
  border: 1px solid black;
  margin: 0px;
  padding: 0px;
  height: 15px;
  background: black;
  color: white;
}
#box3 {
  border: 1px solid black;
  margin: 0px;
  padding: 0px;
  height: 15px;
  background: silver;
}

<div id="box1">
  box box box1
  <div id="box2">box box box2</div>
</div>
<div id="box3">box box box3</div>
```

תרשים 5.16 מציג את התוצאה בדפדפן, ואחריה מוצגת התזוזה של 15 פיקסלים כתמונת פוטושופ. הפעם התגית השנייה זזה 15 פיקסלים ימינה ולמטה באופן מוחלט. כפי שלמדת, הזזה מוחלטת מוציאה את האלמנט מהזרימה הכללית ומחושבת יחסית לאב הקרוב ביותר שהוגדר לו מיקום. במקרה זה ההזזה היא יחסית לתגית div1 שמכילה אותה ושהוגדר לה מיקום יחסי.



## 5.16 תרשים

מיקומים מוחלטים מוציאים את האלמנט מהזרימה הכללית של הדף ויכולים לחפוף אלמנטים אחרים. ניתן לשלוט על "מי מעל מי" בעזרת המאפיין: z-index:10. ככל שהמספר גבוה יותר, כך המיקום הוא בשכבה העליונה יותר.

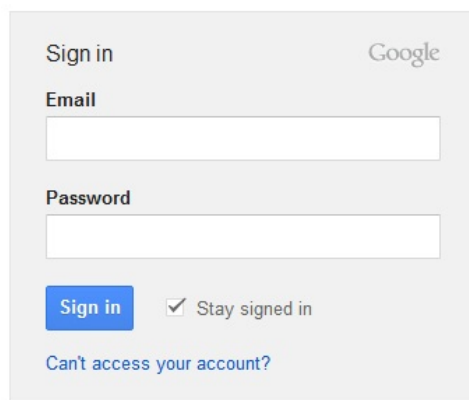
## דוגמה שימושית

רוב דפי HTML5 שנעצב בהמשך הלימוד צריכים להיות מעוצבים ובנויים באופן כזה, שתהיה תגית ראשית ממורכזת באופן אופקי. בתוכה כל התוכן והעיצוב של הדף יהיה ממוקם באופן יחסי לתגית העוטפת. בהמשך הפרק תלמד כיצד למרכז את התגית (הקופסה) הראשית.

## מהו הטופס ומהי מטרתו?

טופס שמוצג באתר אינטרנט או ביישום אינטרנטי, מאפשר למשתמש ליצור תקשורת דו-כיוונית עם האתר. הטופס נועד לאימות, להוספה, לשינוי ולמחיקה של נתונים. פעולות אלו מתבצעות על ידי שליחת נתונים אל השרת, כפי שתראה בהמשך הלימוד בפרק זה.

תרשים 7.1 מציג את טופס ההתחברות ל-Gmail.



The image shows a sign-in form for Gmail. At the top left, it says "Sign in" and at the top right, the "Google" logo. Below the logo, there are two input fields: "Email" and "Password". Under the "Password" field, there is a "Sign in" button and a checkbox labeled "Stay signed in". At the bottom, there is a link that says "Can't access your account?".

### תרשים 7.1

גם חיפוש מידע ב-Google (גוגל) מתבצע על ידי תבנית טופס. ראה תרשים 7.2:



The image shows the Google Israel search page. At the top, the "Google Israel" logo is displayed. Below the logo is a large search input field. Underneath the input field are two buttons: "Google Search" and "I'm Feeling Lucky". At the bottom, there is text that says "Google.co.il offered in: العربية العريية".

### תרשים 7.2

הצטרפות למועדון לקוחות, רכישת מוצרים בחנות מקוונת ופעולות רבות אחרות, נעשות ברוב המקרים על ידי טפסי אינטרנט:

### תרשים 7.3

המטרה של כל הטפסים שראינו ואחרים שמוכרים לך, היא לשלוח את הנתונים אל השרת כדי שיוכל לבדוק את נכונותם, או כדי לשמור אותם בבסיס הנתונים שבשרת. לדוגמה, במהלך ההתחברות ל-Gmail נשלחים שם המשתמש והסיסמה אל השרת, שמשווה אותם לנתונים השמורים בבסיס הנתונים שלו. אם הם נכונים - אפשר להתחבר לחשבון האישי. אם אינם נכונים, תוצג הודעת שגיאה ובה יהיה כתוב שיש טעות בנתונים שהוזנו. במצבים מסוימים, גם פקודות תוכנית javascript בצד הלקוח יכולות לעבד את הנתונים המתקבלים.

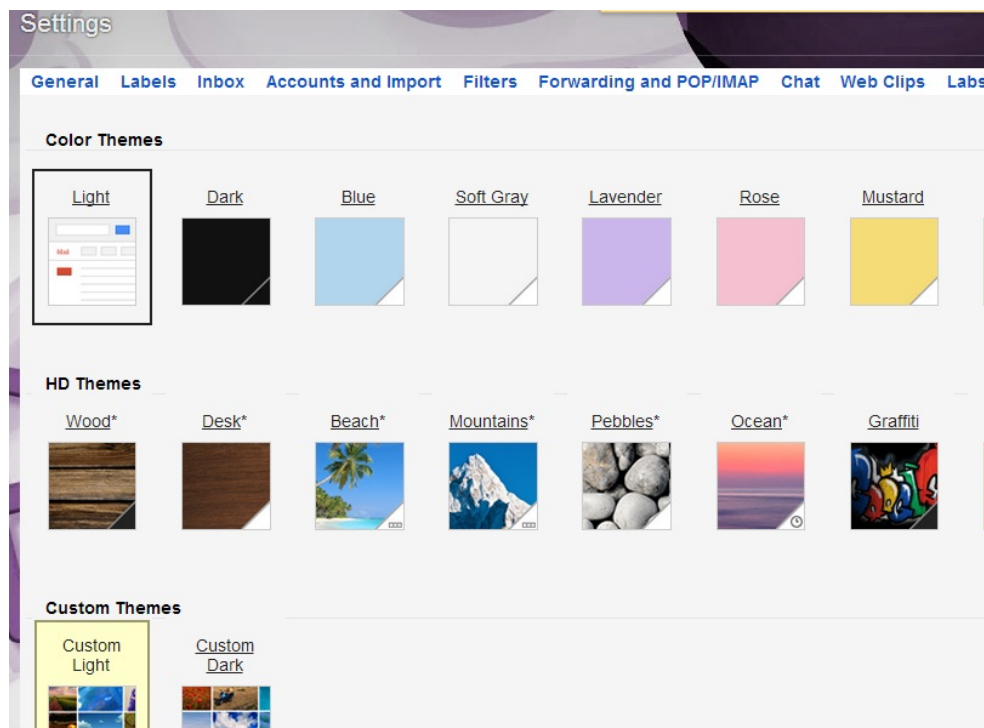
הטופס מורכב מפקדים (Controls). אלה הם שדות בחירה וכתובה שמאפשרים לגולש לבחור ולהזין את הנתונים. בהמשך נכיר את הפקדים ואת אופן שליחת הנתונים לשרת ונראה דוגמאות רבות.

## טכנולוגיה ועיצוב יד ביד

במשך שנים רבות לא היה כמעט שינוי באופן כתיבת טופס ב-HTML, על אף שאופן השימוש באינטרנט השתנה מקצה לקצה. ממשתמש פסיכי שקורא מידע בלבד, הפכנו להיות משתמשים אקטיביים ששולחים מידע, מעדכנים ומשנים אותו. במילים אחרות, בתקופה הראשונה, הגולש קרא תוכן באינטרנט בלבד; בתקופה השנייה הגולש העלה את התוכן ושינה אותו כרצונו (בלוגים, פורומים, הערות למאמרים ועוד).

עם הזמן נולד צורך חדש: להתאים את המידע לצרכים של הגולש. כך הגענו לתקופה השלישית שמתאפיינת בהתאמת המידע לצרכים האישיים של הגולש ובניהול המידע והידע שכמותם ועוצמתם הולכת ועולה ברשת האינטרנט. קיימים כיום ממשקים שלגולש יש אפשרות לשנות את צבע הרקע שלהם, את הגאדג'טים (תוכניות עזר שונות) שיופיעו בהם, את הפונקציונליות שלהם ועוד.

לדוגמה, ב-Gmail אתה יכול להגדיר תמונת רקע שתופיע בשקיפות עדינה מתחת למיילים שלך.

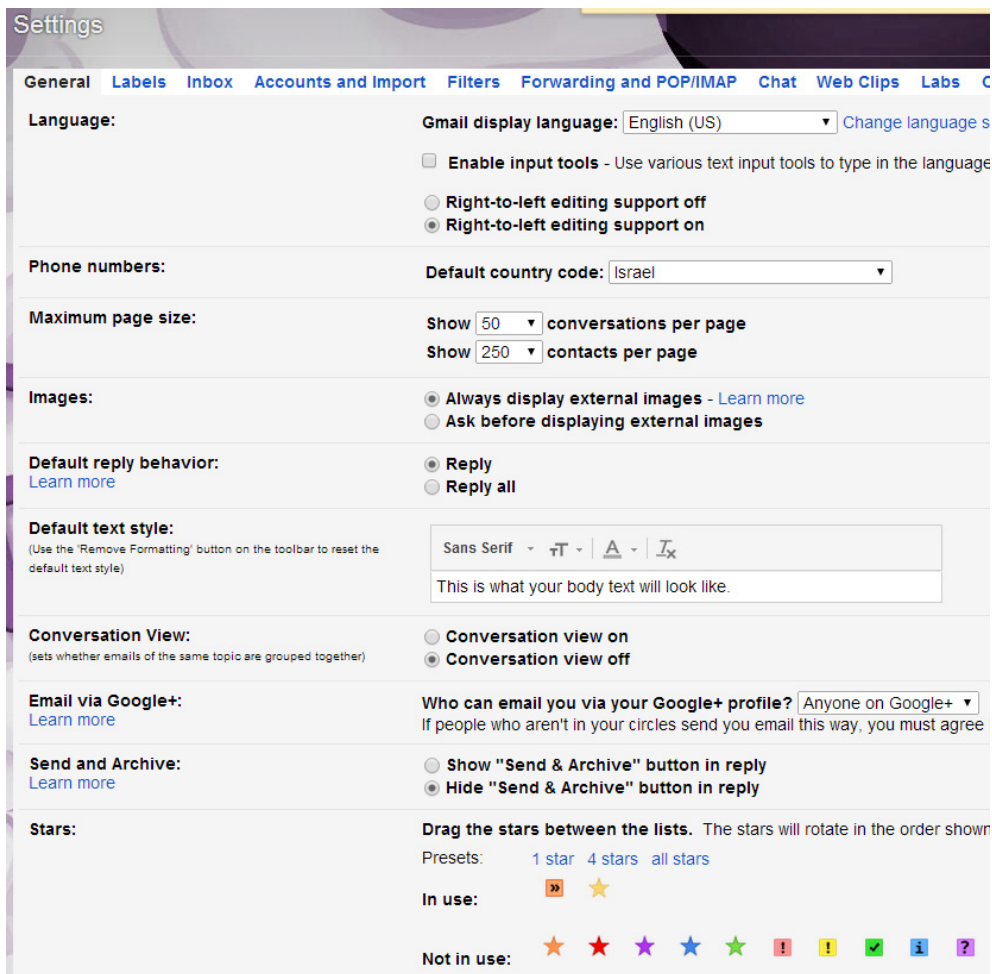


#### תרשים 7.4

אפשרויות ההתאמה האישית (Customization) של מערכות ואפליקציות, נעשות בדרך כלל על ידי טפסים. תרשים 7.5 מציג חלק מהאפשרויות של המשתמש ב-Gmail.

אלה הם רק חלק מהשינויים שעוברים אתרי האינטרנט, הממשקים והיישומים. גישות אלו לשימוש באינטרנט ידועות בשמות web2 ו-web3. מעבר להעשרה הכללית של הידע ולהיכרות עם מונחים מהעולם הזוהר של אפליקציות ואתרים אינטראקטיביים, הבנת צרכי הגולש בימינו חשובה למעצב האתר כדי שיוכל לספק אתרי אינטרנט איכותיים, עדכניים מבחינת תפישתם ועכשוויים.

הגישה כיום גורסת, שכדי להפיק אתר אינטרנט, מערכת מידע אינטרנטית או אפליקציית web מקצועיים ברמה בינלאומית, על המתכנתים להבין ולהכיר את עולם העיצוב, ועל המעצבים להבין ולהכיר את יכולות הטכנולוגיה. כשהמתכנתים והמעצבים מבינים אלה את אלה, מתקבלים האתרים האינטראקטיביים הטובים ביותר!



## תרשים 7.5

### החשיבות של תכנון ועיצוב הטפסים

הגולש מזין תוכן, משנה תוכן, מסמן ובוחר, וכך נוצר קשר (אינטראקציה) בין הגולש לבין אתר האינטרנט. קשר זה נוצר לרוב על ידי שימוש בטפסים.

טפסים באתר האינטרנט הינם הדרך היעילה והנפוצה ביותר ליצירת קשר עם גולש. הקשר הזה יכול להיות באמצעות דף "צור קשר", בטופס "רישום למועדון הלקוחות" או בטופס "פרטים", כשהגולש רוצה לרכוש מוצר הנמכר באתר ועליו להשאיר לשם כך את פרטיו האישיים ופרטי כרטיס האשראי שלו. תכנון, עיצוב ובניית טופס המשרת את המטרה שלשמה נבנה הטופס הכרחי להשגת אותה המטרה.



הדפדפנים השונים תומכים בקבצים מסוג mp3 ו-ogg. ניתן להמיר את קבצי mp3 לקבצי ogg באמצעות תוכנות לעריכת מוסיקה ובאמצעות כלים שנמצאים ברשת, כמו <http://audio.online-convert.com/convert-to-ogg>. קוד `exercise_10.4.html`:

```
<audio controls>
  <source src="sounds/1.mp3" type="audio/mp4">
  <source src="sounds/1.ogv" type="audio/ogg">
  no support
</audio>
```

קוד Javascript מאפשר שליטה על אובייקט שמע (אודיו) בדומה לשליטה על אובייקט וידאו.

## התגית Canvas

### הכרת התגית, דפדפנים תומכים, מטרה

התגית **Canvas** (קנבס) והטכנולוגיה הנלווית לה הן אחד החידושים המשמעותיים ביותר ב-HTML5. רבות נטען ש-HTML5 מחליפה את השימוש בטכנולוגיית Flash לתכנות משחקים, אנימציות, גרפים ודיאגרמות אנימטיביות ועוד. ואמנם, אחת הסיבות לטענה זו היא השימוש בתגית `<canvas>`.

התגית קנבס `<canvas>...</canvas>` מספקת משטח עבודה (ומכאן שמה) לאיור צורות, תמונות, טקסטים ועוד.

ההצלחות הבולטות של הקנבס:

אינטראקטיביות - הקנבס מגיב לאירועים שהמשתמש יוזם, כמו מעבר עכבר, לחיצת עכבר ועוד.

אנימטיביות - ביחד עם האינטראקטיביות, אלו שתי התכונות שמאפשרות לתכנת משחקים בקנבס במקום בפלאש. כל צורה שאתה מצייר על הקנבס אתה יכול להנפיש.

תמיכה נרחבת בדפדפנים - אמנם תראה כיצד לזהות אם הדפדפן תומך בקנבס, אבל כיום רוב הדפדפנים תומכים בו.

פופולריות - קהילות מתכנתים מציגות לראווה דוגמאות מורכבות, מעוצבות, מדהימות. אני ממליצה לך להתרשם מהכמות העצומה של דוגמאות מרהיבות של מתכנתים ומעצבים באתר <http://www.chromeexperiments.com>.

במסגרת פרק זה נעסוק בגרפיקה דו-ממדית, אבל כדאי לדעת שקיימת תמיכה נרחבת גם בגרפיקה תלת-ממדית, כפי שיוסבר בהמשך. דפדפנים שאינם תומכים בתגית `<canvas>`,

מציגים טקסט מחליף כפי שנקבע להם. ניתן להגדיר יותר מתגית <canvas> אחת בדף והתגיות עשויות לחפוף זו את זו.

חשוב לדעת שכל מה שמאויר, משורטט או נכתב - ולמעשה מוצג במשטח העבודה קנבס - לא נשמר בשום מקום על ידי הדפדפן. כלומר, אם אתה מצייר משהו בקנבס וסוגר את הדפדפן, תצטרך לערוך את הציור מחדש בפעם הבאה שתוציא להציגו. זאת בניגוד לצורה שמוגדרת באובייקט SVG, עליו לא נרחיב בפרק זה. לכן, טכנולוגיית קנבס מתאימה להצגה חד-פעמית של גרפיקה, ואינה מתאימה כאשר צריך להשתמש בתוכן המוצג שוב ושוב. התוכן שמוצג במשטח העבודה אינו חלק מהדף, אלא נוצר על ידי הקוד כתצוגה חולפת, on the fly. בכל פעם שתפתח את העמוד יופעל הקוד שמצייר על הקנבס, אך לא תישמר הנקודה האחרונה שצוירה. הקוד יתחיל לצייר מההתחלה.

## הוספת התגית למסמך

כל משטחי העבודה שקופים בתחילה. ניתן להגדיר יותר ממשטח עבודה אחד בדף והמשטחים יכולים לחפוף זה את זה. כמתחיל, הגדר תחילה משטח עבודה אחד ופעל בו, ואחר כך הוסף משטח ופעל גם בו בדרך דומה.

אם לא תגדיר את גודל משטח העבודה, הוא יוצג בגודל ברירת המחדל: רוחב 300 פיקסלים, גובה 150 פיקסלים.

להלן דוגמת שימוש בתגית - קוד canvas\_10.1.html:

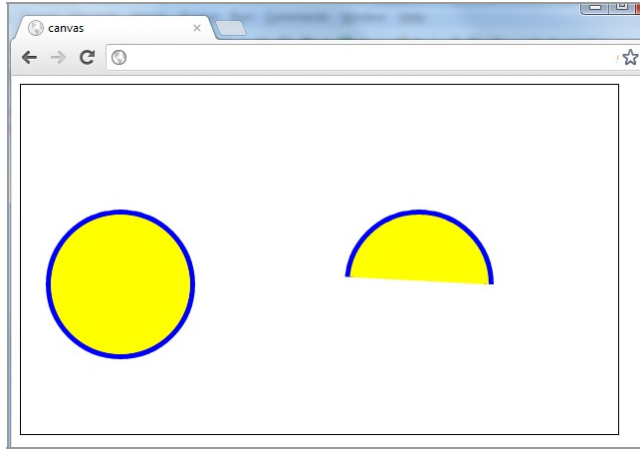
```
<canvas>
  Fallback Content
</canvas>
```

שים לב, מספרי הקודים בחלק זה של הפרק ממוספרים בנפרד עם המילה canvas בהתחלה, לצורך ההפרדה בין ההסבר על סרטונים ואודיו לבין החלק שעוסק בתגית <canvas>.

הדפדפן קורא את התגית, ומציג את מה שכתוב בקוד, כפי שתלמד בהמשך. אם הדפדפן מגרסאות קודמות ואינו תומך בתגית קנבס, יוצג בו הטקסט הכתוב בין תגית הפתיחה לבין תגית הסגירה. בדוגמה הקודמת canvas\_10.1.html יהיה כתוב Fallback Content.

תגית זו אינה מציגה דבר והדף נשאר לבן וריק. לכן, נוסיף מעט הגדרות עיצוב ב-CSS. להזכירך, כשיש ב-CSS הגדרות עיצוב שמתייחסות לתגית באופן כללי, כותבים תחילה את שם התגית ולאחריה את הגדרות העיצוב שמשויכות אליה:

```
canvas
{
  border: solid black 1px;
}
```



תרשים 10.17

## עקומות

כדי לצייר עקומות עליך להשתמש בפונקציה `bezierCurveTo()`.

## הכרת המצב (state) של משטח העבודה

לכל אובייקט `context` (אובייקט הקשר) של קנבס יש `drawing state` (מצב שרטוט) שהקוד יכול לנהל. ניתן לשמור את המצב הנוכחי או לשחזר את המצב הקודם שהוגדר. מצבי משטח העבודה השונים נשמרים במחסנית נתונים, מכיוון שבכל פעם ששומרים מצב חדש `(state)`, הוא מוכנס למחסנית וכשמשחזרים הוא מוקפץ לפי סדר LIFO (אחרון נכנס, ראשון יוצא).

להלן הפרמטרים שנשמרים עבור כל מצב `(state)`:

- ערך נוכחי של הפרמטרים `lineWidth`, `strokeStyle`, `fillStyle`, `lineCap` ואחרים.
- טבלת ההמרה (Transformation matrix) הנוכחית (לא נרחיב נושא זה).
- אזור הלכידה (Clipping region) הנוכחי (לא נרחיב נושא זה).

הסיבות לשמירת המצב של משטח העבודה: בדרך כלל פועלים עם אובייקט בעל הגדרות עיצוב מסוימות ומשכפלים אותו, מזיזים אותו וכו'. שמירת מצב מאפשר לבצע בו שינוי עיצוב, וגם לחזור לעיצוב המקורי.

השלבים בשמירה ובשחזור מצב ההקשר (context):

1. שמירת מצב על ידי קריאה לפונקציה `.context.save();`
2. ביצוע שינויים.
3. שחזור מצב על ידי קריאה לפונקציה `.context.restore();`

בדוגמה זו (קוד `canvas_10.9.html`) נצייר שלושה קווים:

1. קו שחור.
2. קו אפור.
3. קו ללא הגדרות עיצוב.

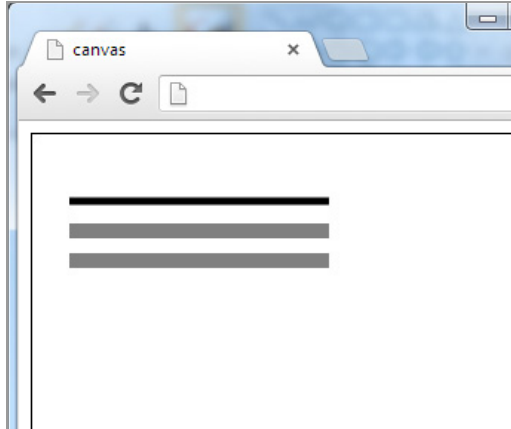
שים לב, שאנו מגדירים בהתחלה קו בצבע שחור ובעובי 5. לאחר ציור הקו הראשון אנו משנים את הגדרת עיצוב הקו. אנו מגדירים קו בצבע אפור ובעובי 10. לאחר ציור הקו השני אנו מציירים קו שלישי, אך לא מגדירים לו צבע ועובי. הבה ננסה לנחש כיצד ייראה הקו השלישי: כמו הראשון או כמו השני?

```
function go_draw(ctx) {
  //init drawing parameters
  ctx.strokeStyle = "black";
  ctx.lineWidth = 5;

  //1. draw first line
  ctx.moveTo(25,45);
  ctx.lineTo(200,45);
  ctx.stroke();

  //2. draw second line
  ctx.beginPath();
  ctx.strokeStyle = "grey";
  ctx.lineWidth = 10;
  ctx.moveTo(25,65);
  ctx.lineTo(200,65);
  ctx.stroke();

  //3. draw third line
  ctx.beginPath();
  ctx.moveTo(25,85);
  ctx.lineTo(200,85);
  ctx.stroke();
}
```



### תרשים 10.18

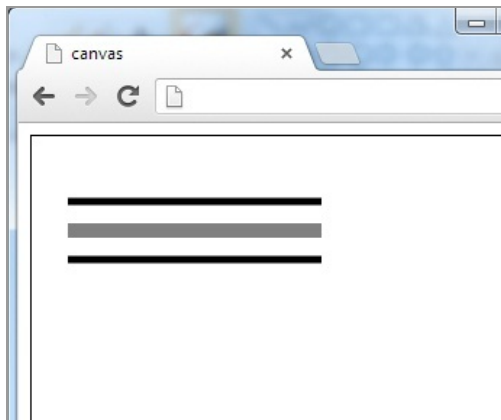
הקו השלישי זהה לקו השני. נעשה כעת את השינוי הבא:

- נשמור את המצב לאחר ציור הקו הראשון.
- נשחזר את המצב לאחר ציור הקו השני.

הנה הקוד - canvas\_10.9.html :

```
function go_draw(ctx) {  
    //init drawing parameters  
    ctx.strokeStyle = "black";  
    ctx.lineWidth = 5;  
  
    //1. draw first line  
    ctx.moveTo(25,45);  
    ctx.lineTo(200,45);  
    ctx.stroke();  
  
    ctx.save();  
  
    //2. draw second line  
    ctx.beginPath();  
    ctx.strokeStyle = "grey";  
    ctx.lineWidth = 10;  
    ctx.moveTo(25,65);  
    ctx.lineTo(200,65);  
    ctx.stroke();  
  
    ctx.restore();
```

```
//3. draw third line
ctx.beginPath();
ctx.moveTo(25,85);
ctx.lineTo(200,85);
ctx.stroke();
}
```



### תרשים 10.19

לאחר שחזור המצב, הקו השלישי קיבל את הגדרות הקו הראשון.

## יצירת צל

נצייר כעת מספר טקסטים כפי שמתואר בסעיף הקודם, ונוסיף להם צל. ליצירת צל נוכל להיעזר בכמה פונקציות מובנות:

**shadowColor** - צבע הצל.

**shadowBlur** - הטשטוש של שולי הצל. ערך זה חייב להיות גדול מ-0.

**shadowOffsetX** - הטיה על ציר X (ברירת מחדל 0). מספר שלילי יגרום להופעת הצל משמאל ומעל האובייקט.

**shadowOffsetY** - הטיה על ציר Y (ברירת מחדל 0). מספר שלילי יגרום להופעת הצל משמאל ומעל האובייקט.

כל אובייקט שכתוב בקוד אחרי הגדרות הצל - יושפע מהגדרות אלו.

קוד `canvas_10.12.html` מציג תוכנית שלמה, אשר מציירת צל למלבן שציירנו בדוגמה 6 ולשורת טקסט שחור:

```

function go_draw(ctx) {
    var canv = document.getElementById("canv");
    canv.width = 600;
    canv.height = 400;

    ctx.shadowColor = "#666666";
    ctx.shadowBlur = 7;
    ctx.shadowOffsetX = 8;
    ctx.shadowOffsetY = 15;

    ctx.strokeStyle = "black";
    ctx.lineWidth = 8;
    ctx.fillStyle = "yellow";

    ctx.strokeRect (ctx.canvas.width/2, ctx.canvas.height/2, 30
, 30);

    ctx.fillRect (ctx.canvas.width/2, ctx.canvas.height/2, 30, 3
0);

    var dynamic_txt = "hello, my name is Yonit";
    ctx.fillStyle = "black";
    ctx.font = "42px Georgia";
    ctx.fillText (dynamic_txt, 20, 320); //defalut settings
}

window.onload = function() {
    if (!!document.createElement('canvas').getContext)
    go_draw (document.getElementById("canv").getContext("2d")
);
}

```



## תרשים 10.20

## היסודות הדרושים לאתר רספונסיבי

הבסיס לבניית אתר רספונסיבי אשר מותאם למכשיר מובייל כולל כתיבת קוד HTML5, כפי שלמדנו בפרקים השונים של ספר זה. הקוד צריך להיות "נקי", תכליתי, בעל משמעות, מינימליסטי ותקני, כפי שהודגש במהלך הלימוד בספר. בנוסף לכך יש להקפיד שטפסים ישתמשו בסוגי שדות שמותאמים אוטומטית למובייל, כמו רשימות נגללות שמותאמות לטלפונים.

בנוסף לאלה, CSS media queries, טכניקה שתכיר בפרק זה, תאפשר לך לזהות רוחבי מסך ורזולוציות ולהציג פריסה (layout) וטיפוגרפיה (עיצוב הגופנים, הפסקאות והטקסטים בדף) תואמות, על ידי קריאת classes שונים ב-CSS.

CSS3 transitions מאפשר ליצור אנימציות ללא שימוש בקוד נוסף. כלי התכנות המקובלים JQuery ו-Javascript מאפשרים להוסיף אינטראקטיביות, כפי שלמדנו בפרק 9.

## הגדרת Viewport (מעין קנה מידה)

חלון וירטואלי, או מלבן דו-ממדי, שדרכו רואים בקטן את האתר במובייל. הדבר דומה לצפייה בקטן נוף דרך חלון המכונית, או דרך עינית המצלמה.

אם אתר מתוכנן לרוחב כלשהו וה- viewpoint של המובייל הוא 320 פיקסלים, אזי יותאם כל גודל האתר לקנה המידה הרלוונטי, כפי שתוכל לראות בתרשים 12.14. ברירת המחדל היא, שבהתחלה רואה הגולש את כל הדף, ואז הוא יכול להקטין או להגדיל אזור מסוים כדי לקרוא.

## איך שולטים על ה-viewport?

השליטה פשוטה ומיושמת על ידי תגית meta בראש הדף, ב- <head> ובה תופיע הגדרת ה-Viewport:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

בסעיפים הבאים תמצא הסבר לשורת קוד זו, אך שמור אותה - היא תהיה משמעותית ביותר עבור האתר או האפליקציה הרספונסיבית שלך. אני ממליצה לך להתרגל להוסיף שורת קוד זו בתגית <head> בכל הקבצים שאמורים לתמוך במובייל. כך, ביחד עם הגדרות עיצוב נפרדות למסך רחב ולמסך צר, שתלמד להגדיר בהמשך הפרק, תקבל פתרון רספונסיבי ראוי.





**לחצו וגלו את הסמארטפון הטוב בעולם**
  
 Sony
   
 make.believe
   
 Xperia Z

**נולני**
  
 ריאליטי טירונות
   
 ynet מציג: גולני - ריאליטי טירונות

מועלות ynet ליוזם במשך חצי שנה את טירונות גולני - 51 ומקדמים - מהרמלס הראשון של המפגש עם הרצב והסמל ועד להשבעה בכותל. היינו שם ימים, כיום ובלילה, בתיעוד ראשון מסוגו של הכשרת לחמים בגולני. לכובר יום העצמאות, צפו בפרק הראשון בסדרה לסיפור החיל...

**לכל המבקרים**
  
 16.04.13
   
 פרס לחיילים המצטיינים
   
 16/04/2013 10:06
   
 מחשש לעומס יפתחו רק
   
 16/04/2013 09:53
   
 המשטרה מבקשת את
   
 16/04/2013 09:37

**"אנחנו צריכים את החרדים"**
  
 המוסלמי ג'וז מכה לחרדים, מרצה מההרדנה בינה
   
 ומבחינה "בכיסותן צריך להשקיע" ריאון (חדשות)

**בן 8 שודד את אביו - נהרג**
  
 מותו רצוף הוא אחד שילאשת רחמי הפיגוע בבוסטון,
   
 הראשון בארצות הברית - 17 במגז אמש (חדשות)

**תשעה זמרים, תשעה שירים**
  
 פזן פזם חוזרת לזרן, לראו נרקיס מחד בקמלן; זמרים
   
 בחרו במקום הכי ישראלי - ושור לם (תרבות)

**למה ישראלים לא מנמסים**
  
 למה ישראלים אינם מנמסים
   
 למה ישראלים מקרים בברים
   
 למה ישראלים מקרים בברים
   
**גוגל, בא לחומס?**
  
 מה גרם לחומס?
   
 מה גרם לחומס?
   
 מה גרם לחומס?
   
**לא רק סמסא**
  
 50 המסאטור אינם
   
 שצמיים להפוך להצלחה
   
 הישראלית הבאה

**הפיגוע בבוסטון**
  
 ישראל חוגגת את יום העצמאות ה-65
   
 הם פריז למגל

מה אתה מחפש
   
 כבת חנוך שלי?
   
 דרוש מאגר
   
 בוט או חיוני
   
 ימיני גיטי
   
 אחר...
   
 מצא אהבה >>

פנסים טילים
   
 30% הנחה בחקיקה
   
 בעלי מקצוע
   
 דרושים
   
 ZAP
   
 אמנות לבית
   
 ZER4U

**Emalon.co.il**
  
 חמשה ישראליים במחיר מושלם
   
 מיוחד ynet.co.il

**מוליך במחיר קטן**
  
 דוד עם מחר מנגן אלקטרוניקה,
   
 מצלמה ועוד - 199 ש"ח בלבד!

**דיסקונט**
  
**digital**
  
**גוגל פועלים**
  
**מזרחי טפחות**
  
**b14.co.il**
  
 כל היקום, כל הזמנים
   
**סלנוואט**
  
 לישג, רשתות ורוב
   
 מפיות רוב ישראלי דרך
   
**לפתיחת סלולג**
  
**המבצעים של**
  
**סופר-פארם <<**
  
 בית חושפיה
   
 ובישום

**תרשים 12.14**

## הפרמטרים השולטים על ה-Viewport

**width**: בהגדרת meta כותבים `width=320`. בהגדרות CSS כותבים `width:320px`. שים לב, שיש הגדרת פיקסלים רק כשמגדירים את ה-viewport ב-CSS: `320px`. לעומת הגדרה ללא ציון px כשהוא מוגדר בתגית meta: `width=320`.

**device-width**: מגדיר את הרוחב המלא של הדפדפן במובייל, כדי לא להגביל לרוחב מסוים בפיקסלים (משמע, width יכול להיות רוחב הדפדפן במובייל, כשה-device-width זה הרוחב של המכשיר).

**height**: בדומה להגדרת הרוחב אפשר להגדיר גם גובה, אם כי לא עושים בה שימוש רב. **initial-scale**: הגדרת הזום ההתחלתי. בזמן התצוגה הגולש יכול לשנות את הזום בשיטות הרגילות לשינוי הזום במכשיר.

לדוגמה, `initial-scale=1` משמע, הצג את הדף ב-100%

לעומת, `initial-scale=2` משמע, יציג את הדף ב-200%

ההגדרה `width=device-width, initial-scale=1` הינה המומלצת ברוב המקרים.

קוד 12.1 להלן מציג הגדרה של הכפלת הזום והתאמה לרוחב של 640 פיקסלים. נסה לשחק עם ההגדרות האלו, לשנות את הרוחב, להגדיר `initial-scale`, `device-width` בפיקסלים או באחוזים, ולראות כיצד זה משפיע על התצוגה במכשיר המובייל:

```
<!doctype html>
<html>
  <head>
    <title>viewport</title>
    <meta name="viewport" content="initial-scale=2,
width=640">
  </head>
  <body>
    <h1>VIEWPORT's most important parameter: INITIAL-
SCALE</h1>
    <h2>viewport set by meta to initial-scale=2,
width=640</h2>
    <p style="background:yellow;">Lets try to change the
viewport of this page by changing the meta tag or the css
@viewport option.</p>
  </body>
</html>
```

### קוד 12.1

## פרמטרים נוספים של Viewport

הסבר	CSS	meta
מגביל את הזום שיכול לעשות הגולש	min-zoom:1;	minimum-scale=1
מגביל את הזום שיכול לעשות הגולש	max-zoom:2;	maximum-scale=2
יכול למנוע יכולת זום של הגולש. פחות משתמשים בזה באתרי אינטרנט	user-zoom:fixed;    או user-zoom:zoom;	user-scalable=no    או user-scalable=yes

## density - צפיפות פיקסלים במסך

לא לבלבל בין המושג 'צפיפות' לבין המושג 'רזולוציה'. רזולוציה היא כמות הפיקסלים שניתן להכניס במסך. ככל שהמסכים חדשים יותר כך הם תומכים ביותר פיקסלים.

דוגמה:

אייפון 3S תומך ב- 320 פיקסלים רוחב על 480 גובה.

אייפון 4S תומך ב- 640 פיקסלים רוחב על 960 גובה.

למרות השוני של מספר הפיקסלים במסך, גודל המסך זהה בשני המכשירים האלה.

כדי שלמכשיר 4S תהיה רזולוציה גבוהה יותר, עליו להציג יותר פיקסלים על המסך. כלומר, כאשר מגדירים - height:8px;width:4px - האובייקט שמוצג יראה קטן במסך שיש בו יותר פיקסלים, כפי שניתן להבחין בתרשים 12.15:

## שימוש במסגרת קוד - framework

כפי שראית, יצירת דפים רספונסיביים מחייבת מחשבה ועבודה רבה. אמנם, אין זה כמו תכנון אתר נפרד למובייל, אך עדיין יש פה אתגר לא פשוט. לכן, מומלץ להיעזר במסגרת קוד (framework). מסגרות קוד קיימות ברשת למכביר, והן כוללות קבצי HTML5, Javascript ו-CSS אשר מאפשרות, בהתחשב בתכונות והאפשרויות שהן מספקות, לבנות מערכות ואתרים במהירות.

דוגמה מקובלת מאוד, שמסייעת בהתאמה למסך הצר ולמסך הרחב, תוך התייחסות להתנהגות "של יישום" לעומת התנהגות של מסך (מעברי עכבר וכד') היא המסגרת של JQuery mobile.

מסגרת הקוד המקובלת של JQuery mobile מאפשרת ליצור בנחות רבה אנימציה של תזוזת דפים לצדדים, כפי שמקובל היום במכשירים הניידים הפועלים במגע אצבע, ליצור רשימות (lists) ותיבות שיחה צצות (lightbox).

על ידי CSS ועל ידי JQuery mobile ניתן ליצור דפים שמגיבים היטב למסכים צרים. תוכל להשתמש בלחצנים, ברשימות, בתיבות סימון (checkboxes) ובפקדים אחרים, כדי להציג טפסים המותאמים לפעולת האצבע של הגולש ולהתנהגות המשתמש במכשיר נייד.

אחד ה-frameworks המקובלים והידועים ביותר, המאפשר תמיכה מלאה ברספונסיביות האתר, ומציע סגנונות מוכנים מראש עבור מתכנתים, הוא ה-Bootstrap.

## Bootstrap

המילה bootstrap מתייחסת בעולם התוכנה לאתחול של מערכת. משמע, להוראות ראשוניות שיש להגדיר כדי שאפליקציה, תוכנה, אתר או מערכת יוכלו להתחיל לפעול באופן תקין.

מסגרת הקוד Bootstrap Framework מתייחסת להגדרות הכתובות בקוד פתוח, קוד הניתן לראות אותו ברשת, והמאפשרות אתחול של אתר אינטרנט באופן שיקל על בוני האתר לבנות אותו.

ההגדרות כוללות תגיות html והגדרות css שהשימוש בהן יאפשר לך לבנות אתר אינטרנט רספונסיבי, בעל נראות מקצועית על פי המוסכמות המקובלות היום. ההגדרות מתייחסות לרספונסיביות, למבנה עמוד (גריד), לפונטים, לצבעים, לכפתורים, לתפריטי ניווט ולעוד אלמנטים הנדרשים בכל אתר ואתר.

## היתרונות בשימוש ב- Bootstrap

היתרון הראשון הבולט הוא חוכמת ההמונים. המשמעות היא שמתכנתים רבים לקחו חלק בקהילת Bootstrap ותרמו לקוד. התוצאה היא אלפי שורות של קוד מוכן עבורך לשימוש. קוד בטוח לשימוש, מכיון שמיליוני בוני אתרים ברחבי העולם עשו ועושים בו שימוש יומיומי ותורמים לשיפורו המתמיד.

יתרון נוסף הוא עצם השימוש בעיצוב ובפריסת עמוד (גריד) כשאתה בונה אתר ללא מעצב גרפי. התוצאה מרשימה ומקצועית, והשימוש במוסכמות מבחינת הנראות משפר ומשדרג את איכות התוצר הסופי שלך.

יתרון שלישי הוא התמיכה הנרחבת בשינויים בקוד והתאמות אישיות לצרכים של אתר האינטרנט שלך. תכונה זו נקראת customization לקוד.

יתרון רביעי נובע מהעובדה ששימוש בקוד מוכן זה מאפשר לך תכנות מהיר ובטוח.

## החסרונות בשימוש ב- Bootstrap

חסרון ראשון הוא שצריך לדעת להשתמש רק בקוד שרלוונטי לפרויקט שלך. Bootstrap מכילה אלפי שורות קוד ועליך להכיר ולהתמצא בכך, על מנת שתוכל לדעת במה לעשות שימוש. עליך ללמוד את הסביבה החדשה וזהו חיסרון בולט. כמוכן שמצד שני, לאחר שתלמד ותכיר את Bootstrap, הדבר ישמש אותך לכל אתר נוסף שתבנה.

חסרון נוסף הוא כמויות הקוד שלא צריך. יש הרבה קטעי קוד והגדרות עיצוב שלא צריך ועליך לדעת לא להשתמש בהם. חלקם מגיעים מוטמעים בקובץ שאתה תקשר לאתר שלך, כפי שתלמד בהמשך, ללא יכולת להוריד ממנו את קטעי הקוד שלא צריך. התוצאה היא קוד שיכול להכביד על האתר שלא לצורך.

## הטמעת הקוד של Bootstrap באתר האינטרנט שלך

בדומה להטמעת התמיכה ב-JQuery, כך נדרשת הוספת התמיכה בהגדרות של Bootstrap. לשם כך, קיימות שתי אפשרויות. אפשרות ראשונה היא הורדת קבצי Bootstrap לספריית הפרויקט שלך. היכנס לאתר האינטרנט של Bootstrap בכתובת: <http://getbootstrap.com>, לחץ על הכפתור או הקישור "הורדה" והורד את הקבצים. עדיף להוריד קבצים מוקטנים שהסיומת שלהם היא min. את הקבצים שהורדת העבר לספריית includes באתר שלך. בהמשך לכך, הוסף קישורים לקבצי קוד אלו על פי ההנחיות באתר של Bootstrap, בעזרת תגיות `<link>` ב- `<head>` ו- `<script>` לפני סוף תגית ה- `<body>`.

ברחבי האינטרנט קיימים העתקים רבים של קובץ התמיכה ב-Bootstrap. העתקים אלו יכולים להיות אמינים ומטרתם לעזור להטמיע ספריות ומסגרות JavaScript ו-CSS ללא

צורך להוריד את הקבצים למחשב. לכן, האפשרות השנייה העומדת לפניך היא לעשות שימוש באחד מהעתקים אלו. לידיעתך, גם את ספריית JQuery ניתן להטמיע בדרך זו.

לשם כך, העתק את הקוד המוצג באתר של Bootstrap תחת הכותרת BootstrapCDN ישירות לתוך תגית <head> בכל עמוד באתר האינטרנט שלך שבו אתה רוצה לעשות שימוש ב-Bootstrap.

להלן דוגמה של קוד, עם הטמעת Bootstrap:

```
3 <head>
4 <title>Shenkar 2018 - FINAL PROJECTS</title>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=
7
8 <!-- Bootstrap CSS -->
9 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/b
10 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
11 <link href="includes/style.css" rel="stylesheet" type="text/css">
12 </head>
```

תוכל לראות שבשורות 9, 10 מוטמעים הקישורים לקובץ ה-CSS ולקובץ ה-JavaScript של Bootstrap. הקבצים המקוריים, בהן מוגדרות הפקודות והפונקציות, מאוחסן בשרת ש-Bootstrap עושה בו שימוש. במקרה שאתה מטמיע את רכיב Bootstrap באופן כזה, אין צורך להוריד את הקבצים למחשב שלך. שים לב לשורה 11 - הקישור לקובץ ה-CSS שלך חייב להיות מוגדר לאחר הגדרת קובץ ה-CSS של Bootstrap.

## פריסת העמוד, הגריד של Bootstrap

בפרק 5 למדת כיצד למרכז את האתר. Bootstrap מאפשרת להשתמש בתגית wrapper רספונסיבי משלה. התגית של Bootstrap נקראת container והיא ממרכזת את האתר כפי שלמדת.

לאחר הטמעת רכיב Bootstrap באתר שלך כראוי, השלב הראשון הוא לרשום את הקוד הבא במקום להשתמש ב-wrapper שלך.

```
<div class="container">
```

```
</div>
```

השלב השני בעבודה עם Bootstrap הוא להשתמש בשיטת ה"גריד" לבניית פריסת העמוד. לצורך הבנת שיטת הגריד, נבין תחילה את המושג המרכזי "גריד".

ראה בתרשים 12.22 את האתר הרשמי של Bootstrap, כשבשכבה מעליו מוצגים שנים עשר פסים מלמעלה למטה. כל הפסים ברוחב זהה, עם מרווח שווה ביניהם ושוליים שווים