

## הקדמה

אני חושב שאחד הצעדים הכי גדולים וחשובים בתהליך הלימוד של מתכנת הוא הצעד הראשון. יש פער מנטלי מאוד גדול בין הרהורים ומחשבות כנות לגבי האם אני רוצה להיות מתכנת, לבין באמת להשקיע את מלוא זמנך וללמוד את המקצוע הזה. הצעד הראשון הזה מאוד מפחיד ותמיד נראה גדול יותר מכפי שהוא באמת. כדי להקל את התהליך הראשוני הזה כתבתי את המדריך הזה. המטרה ברורה - לפשט ככל האפשר את תהליך התחלת לימודי התכנות. לכן המדריך הזה כתוב בעברית ומתעסק ביסודות של שפות התכנות כל מסע מתחיל בצעד הראשון, אומרים שזה הצעד הכי חשוב :-)

## מבוא

כמתכנתי front-end אנו כותבים קוד אשר נשלח לדפדפן של המשתמש, אך איננו יודעים באיזה דפדפן הוא משתמש, זה יכול להיות כל אחד מהדפדפנים המוכרים כיום כגון כרום, פיירפוקס, אדג', אקספלורר, סאפארי ועוד. לכן, המטרה שלנו כמתכנתים היא שהדפדפן יבין מה כתבנו ויפרש את הקוד כפי שרצינו.

קיימות שלוש שפות בהן אנו מספרים לדפדפן מה אנו רוצים שיקרה - HTML, CSS ו-JAVASCRIPT. תכנות צד לקוח דורש מאיתנו לדעת היטב את שלושתן. זה נשמע די מטורף שצריך לדעת שלוש שפות תכנות בשביל לבנות אתרים. אך למען האמת, רק JAVASCRIPT היא שפת תכנות, ואילו שתי האחרות הן שפות Markup שלמעשה רק אומרות לדפדפן מה התוכן שאנחנו רוצים שיוצג בדף, באיזה מבנה ובאיזה עיצוב.

במדריך זה נעבור בצורה בסיסית על שלוש השפות הללו, תוך כדי כתיבת דף אינטרנט פשוט שאותו אתם תכתבו ותציגו בדפדפן שלכם. הדף הסופי יראה כך



ויכיל כותרת, קצת טקסט וכפתור שיכול לשנות את הצבע של הטקסט. יאללה בואו נתחיל...

## HTML

באמצעות שפה זו אנו מסבירים לדפדפן את המבנה והתוכן של דף האינטרנט, כאשר אנחנו עושים זאת באמצעות תגיות (tags).

לדוגמא, אם נרצה להגיד לו שיש כאן פיסקה עם הטקסט - "This is some text", אנו נשתמש בתגית p

```
<p>This is some text</p>
```

דוגמא **נוספת** - אם נרצה להגיד לו שיש כאן תמונה אנו נשתמש בתגית img

```

```

שימו לב שכל תגית צריכה להיסגר, באמצעות הסימן - /. כאשר בדוגמא של הפיסקה יש לנו תגית פותחת ותגית סוגרת, **ואילוי** במקרה השני התגית הפותחת של ה-img היא בעלת סגירה עצמית (self-closing). בתגיות אשר אינן יכולות לקבל תוכן אנו סוגרים את התגית כבר בתגית הפותחת.

**המשפט האחרון לא כל כך ברור**

ודאי שמתם לב שבתגיות של התמונה הוספנו src="..." , כל דבר שאנו מוסיפים לתגית הפותחת נקרא "תכונה" (attribute). כך אנו נותנים לדפדפן עוד מידע בנוגע לתגית. לדוגמא בתגית של התמונה

```

```

הסברנו לדפדפן שהמקור לתמונה נמצא בכתובת <http://www.mecoder.co.il/me.jpg>, כלומר שם נמצא הקובץ עצמו.

יש סוגים שונים של תגיות, אך מכיוון שזהו מדריך ראשוני, לא ארחיב על כך יותר מדי.

לסיכום נושא זה, דף HTML בנוי מתגיות. בתגית עצמה יהיה מידע על התגית, ובתוך התגית יהיה התוכן שלה.

כעת, **כשאנו** מבינים את הרעיון של התגיות, בואו נסקור כיצד בנוי דף אינטרנטי, או ליתר דיוק דף HTML.

```
<!doctype html>
<html>
<head>
  <title>MeCoder - The best blog on earth...</title>
</head>
<body>
  <h1>How to learn HTML</h1>
  <div>
    First you need a computer, then install an <span>IDE</span>.
  </div>
```

</body>

</html>

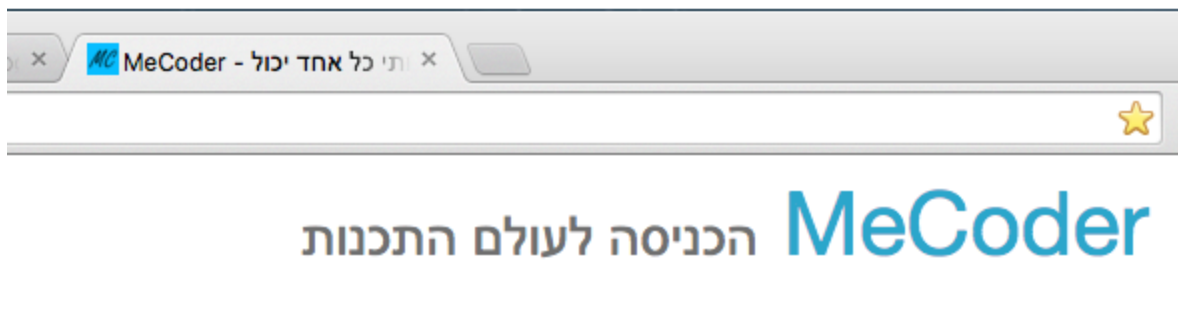
התגית הראשית בדף HTML, נועדה להסביר לדפדפן באיזה גירסה של HTML הדף כתוב. כיום, כמו גם במדריך זה, אנו כותבים בגירסת HTML5, כאשר הדרך לציין זאת היא בצורה הבאה

<!doctype html>

התגית הבאה, אשר למעשה מכילה את כל הדף עצמו, היא תגית html. אין כאן משהו מיוחד, אלא בסך הכל ציון שכל מה שבפנים הוא קוד html. כל התגיות בדף HTML יהיו בתוך התגית הזאת, למעט doctype שצינתי כמה שורות מעל.

בתוך תגית html אפשר לראות שיש שני ילדים (children):

1. התגית הראשונה שתופיעה היא ה-head, אשר מכילה מידע על הדף. בדוגמה המצורפת רשמנו מה הכותרת של הדף באמצעות תגית title. כותרת זו היא הטקסט המופיע על הטאב בדפדפן שלכם. לדוגמה, בבלוג שלי, ה-title הוא "Mecoder - ללמוד תכנות לבד באינטרנט, אם אני הצלחתי כל אחד יכול". כמובן שזהו טקסט ארוך מדי ולכן בתמונה הבאה תראו רק חלק מה-title הזה



2. התגית השנייה היא ה-body, שהיא מכילה את כל התוכן של הדף, כלומר הטקסט, התמונות, קישורים וכו'.

כעת בוא נבחן מה יש בתוך ה-body.

ניתן לראות שבדוגמה שלנו ב-body יש שני children:

1. התגית הראשונה היא h1, שזה קיצור של header1. כלומר זוהי הכותרת הראשית של התוכן בדף. זאת בשונה מה-title, שהיא הכותרת של כל הדף ואינה מופיעה בתוכן של הדף עצמו. מעצם שמה של תגית זו, ניתן להבין שיש עוד סוגים של כותרות עם מספרים יותר גבוהים... נכון, בסך הכל יש שישה headers מ-1 ועד 6. ההבדל ביניהם הוא גם עיצובי, כלומר h1 יראה גדול ובולט יותר מ-h2 וכן הלאה, אך יש להם גם משמעות מבחינת מנועי חיפוש\*.

2. התגית השנייה היא div, שזה קיצור של division. זוהי תגית כללית אשר יכולה להכיל טקסט כפי שמופיע בדוגמה זו, ויכולה להכיל תגיות אחרות כגון p או img או אפילו div אחרים.

בתוך תגית זו מיקמתי תגית span, אשר מציננת תוכן שהוא חלק מהשורה, כלומר לעת עתה אנו לא נראה שום הבדל בין **הטקסט הכתובת (??) מנוח ל-span** לבין זה שבתוכו. נחזור להתעסק באלמנט זה כשנגיע לפרק **שדן ב CSS**.

בשלב זה אני חושב שאנו בשלים להתחיל לכתוב קצת HTML.

כל מה שאתם צריכים זה דפדפן ועורך טקסט, כל השאר זה בונוס. אז תפתחו בבקשה את עורך הטקסט שיש לכם על המחשב, תפתחו קובץ חדש בשם index.html, ותכתבו בתוכו את הקוד הבא

```
<!doctype html>
<html>
<head>
  <title>MeCoder - The best blog on earth...</title>
</head>
<body>
  <h1>How to learn HTML</h1>
  <div>
    First you need a computer, then install an <span>IDE</span>.
  </div>
</body>
</html>
```

לא לשכוח לשמור כמובן (-):  
כעת, גשו לתיקייה ותגררו את הקובץ לדפדפן. זה מה שאתם אמורים לראות

---

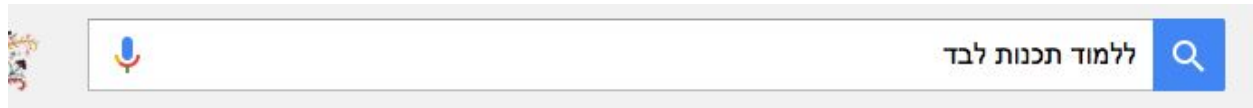
# How to learn HTML

First you need a computer, then install an IDE.

עד כמה שזה יכול להראות טיפשי בשלב כל כך מוקדם ובסיסי, מה שקרה עכשיו זה אחד הקסמים שגרמו לי להתאהב בתכנות front-end, אני מתכוון לפידבק המהיר שאתה מקבל כשאתה כותב קוד. כתבת כמה שורות קוד ומיד התוצאה מופיעה על המסך. קסם!

\* מנועי חיפוש ו-html - השתדלתי לא לגעת בנקודה זו עד כה על מנת להשאיר את הדברים פשוטים, אך ההתעסקות בתגיות כגון h1 מחייבת אותנו להתייחס לנושא זה. מעבר לכך שאנו כותבים html לדפדפנים על מנת שאלו יציגו את התוכן כפי שרצינו, בלא מעט אתרים אנו כותבים את הקוד גם כדי שמנועי החיפוש (google, yahoo, bing...) יבינו מה יש בדף. לדוגמה תגית title מכילה את התוכן שיופיע בכותרת תוצאת החיפוש בה הדף יופיע. ולדוגמה, כשאנו שמים תוכן מסוים בתוך תגית h1 ואילו תוכן אחר בתוך תגית h2, אנו אומרים לדפדפן שהתוכן הראשון (זה שבתוך h1) חשוב יותר מאשר זה שבתוך h2.

לדוגמא, ראו כיצד ה-title של דף הבית שלי מופיע ככותרת תוצאת חיפוש בה הבלוג שלי מופיע



All Videos Images News Maps More ▾ Search tools

About 31,200 results (0.29 seconds)

**MeCoder - ללמוד תכנות לבד באינטרנט, אם אני הצלחתי כל אחד יכול**

[www.mecoder.co.il/](http://www.mecoder.co.il/) ▾ Translate this page

בלוג שמסביר איך אפשר ללמוד תכנות, ומלווה כל מי שרוצה להיות מתכנת אבל לא יודע מאיפה להתחיל.

## CSS

באמצעות שפה זו אנו מעצבים את האלמנטים (תגיות) שהגדרנו באמצעות HTML. יש מספר דרכים להגדיר איזה עיצוב מקושר לאיזה אלמנט, אבל לצורך הלימוד אנו נגדיר את הכללים בתוך קובץ ה-html. לצורך כך אנחנו נפתח תגית style לאחר תגית ה-head שלנו, ובתוכה נכתוב את כללי ה-css שלנו

```
<!doctype html>
<html>
<head>
  <title>MeCoder - The best blog on earth...</title>
</head>
<style>

</style>
...
```

אז איך זה עובד?  
נניח לדוגמא שיש לנו את תגית הוידאו הבאה

```
<video src="http://..." class="pinky-video" />
```

שימו לב שהפעם תגית הוידאו קיבלה תכונה (attribute) נוספת בשם 'class', אשר מאפשר לנו לפנות אליו בקובץ ה-CSS בצורה הבאה:

```
.pinky-video {
  border: 1px solid pink;
}
```

הנקודה המופיעה בתחילת pinky-video אומר לדפדפן שזהו class, כלומר הדפדפן יחפש את כל האלמנטים שיש להם class שהוא pinky-video. לכל האלמנטים הללו הוא יוסיף גבול רציף בצבע ורוד.

על מנת לקבל הבנה יותר רחבה של הדרך בה אנו מחברים בין ה-css ל-html, נשתמש בקוד שכבר כתבנו. רק שהפעם נוסיף לכל תגית אותה נרצה לעצב - class.

```
<!doctype html>
<html>
<head>
  <title>MeCoder - The best blog on earth...</title>
</head>
<style>
```

```

</style>
<body class="whole-document">
  <h1 class="main-header">How to learn HTML</h1>
  <div class="some-text">
    First you need a computer, then install an <span>IDE</span>.
  </div>
</body>
</html>

```

שימו לב שלכל מה שנמצא ב-head לא הוספתי class, זאת משום שתגיות אלו לא מוצגות בדפדפן אלא מהוות רק מידע על התוכן המוצג (meta-data).  
 בואו נתחיל מה-body ונדאג שהוא יתפוס את כל רוחב וגובה המסך וניתן לו קצת צבע כרקע. הוסיפו את הקוד הבא בתוך תגית ה-style

```

.whole-document {
  width: 100%;
  height: 100%;
  background-color: yellow;
}

```

על מנת לראות כיצד קוד שהוספנו השפיע על העיצוב, תרעננו את הדף על ידי לחיצה על כפתור הרפרוש על ידי לחיצה על `ctrl+r`. התוצאה אמורה להראות כך



את הצבע של הרקע הגדרתי באמצעות מילה, אך בדרך כלל אנו נגדיר צבעים באמצעות מספר הקסידמלי (לא להיבהל, זה יותר פשוט ממה שזה נשמע) אשר מייצג את כמות האדום, הכחול והירוק שהצבע מכיל. רק כדי שתכירו ותדעו לזהות מספר שכזה לכשתראו אותו, הוא מיוצג בצורה הבאה - f4ee53

כעת נעצב את הכותרת שלנו. בואו נוסיף לו קו תחתון ונגדיר שצבע הטקסט הוא אדום. נעשה זאת על ידי הוספת הקוד הבא בתוך תגית ה-style

```

.main-header {
  text-decoration: underline;
  color: red;
}

```

}

שוב נשמור ונרענן את הדף, ונראה כיצד העיצוב השתנה



פידבק מהיר כבר אמרתי?...

נותר לנו רק לעצב את האלמנט האחרון. שימו לב שבתוכו יש תגית `span`. בואו נראה כיצד עיצוב ה-`div` ישפיע על עיצוב ה-`span`.  
בואו נגדיל קצת את הטקסט וניתן לו הטייה הצידה בסגנון `italic`, שכולנו מכירים מעורכי תוכן כגון `word`. הוסיפו את הקוד הבא בתוך תגית ה-`style`

```
.some-text {  
  font-size: 24px;  
  font-style: italic;  
}
```

נשמור ונרענן את הדף.

שימו לב שהטקסט שבתוך ה-`span` קיבל גם הוא את הכללים שהגדרנו לאבא שלו, ל-`div`. כלומר, חוקי `css` עוברים בירושה ויש לכך השפעה מאוד גדולה על האופן בו אנו מעצבים דפים.  
אם אתם תוהים מדוע בכלל אנחנו צריכים את תגית ה-`span` הזאת כאן, הרי העיצוב עובר בירושה ולכן אין הבדל ממשי בין אם התגית קיימת או לא, אז אתם צודקים. אבל בואו נראה את אחת הסיבות לכך שכן נרצה להשתמש בתגית זו - במידה ונרצה לשנות את העיצוב של חלק מהטקסט.  
בדוגמא שלנו אנחנו נעצב את התוכן בתגית ה-`span` בצורה קצת שונה, אנחנו נדגיש אותו. אבל רגע... לא הוספנו `class` לתגית זו. איך נגיד לדפדפן שאנחנו רוצים לעצב את תגית ה-`span`?  
אז מסתבר שיש דרך פשוטה ביותר והיא פשוט להגדיר כללי `css` על תגיות במקום על `class`. לדוגמא

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

אם תוסיפו את הקוד הזה, תראו שהכל עובד כפי שרצינו. אבל מסתתרת כאן בעיה! הרגע אמרנו לדפדפן שכל `span` באשר הוא צריך לקבל את הכלל הזה, לא משנה איפה הוא נמצא ולא משנה מי האבא שלו.



כדי לנסות להסביר לו יותר בבירור מי אנו רוצים שיקבל את כללי העיצוב הללו, אנחנו יכולים להגיד לדפדפן שיחיל את כלל העיצוב שכתבנו על כל ה-span שנמצאים בתוך some-text בצורה הבאה

```
.some-text span {  
    font-weight: bold;  
}
```

נשמור ונרענן. כעת הדף שלנו יראה כך



**עכשיו משהבנו באופן בסיסי איך CSS עובד, בואו נעבור ליהלום שבכותרת - Javascript!**  
(המשפט הזה היה מחורבש לשמאל בוורד - אולי בפידיאף זה עבר בסדר)

## JS

אם עד עכשיו התעסקנו בשפות סימון (markup) אשר בסה"כ נותנות מידע על מה צריך להופיע ובאיזה עיצוב, Javascript הינה שפת תכנות. המשמעות העיקרית היא שניתן להגדיר לוגיקה באמצעות הכללים שהשפה נותנת לנו, אבל ל-Javascript יש מאפיין שמבדיל אותה מכל השפות האחרות והוא שהיא "חיה" בתוך הדפדפן וכתוצאה מכך יש לה היכולת לגעת ולהשפיע על דף ה-html עצמו. במדריך הזה אנחנו נתעסק בדיוק בפינה הזו, ולא ניכנס למורכבויות של כתיבת לוגיקה.

לצורך הבנת הפיסקה האחרונה, בואו נכתוב קצת קוד. אנחנו נכתוב פונקציה שכאשר מפעילים אותה היא משנה את הצבע של הטקסט שבתוך ה-div. וכדי להפעיל אותה, אנחנו נוסיף בדף כפתור עליו נוכל ללחוץ.

נתחיל מלהוסיף את הכפתור.

לאחר ה-div אנחנו נוסיף תגית button ובתוכה נשים קצת טקסט

```
<button>Click me!</button>
```

לעת עתה, הכפתור אמנם יופיע בדף שלנו אבל לחיצה עליו לא תעשה כלום. אז בואו ונכתוב את הפונקציה. כדי לכתוב javascript אנחנו נפתח תגית script אשר אומרת לדפדפן שכל מה שנמצא בתוכה הוא קוד JS.

...

```
.some-text span {
  font-weight:bold;
}
</style>

<script>

</script>

<body class="whole-document">
...
```

בתוך תגית ה-script או נכתוב את הפונקציה שלנו, נקרא לה changeColor

```
<script>
  function changeColor () {

  }
</script>
```

כל מה שנמצא בין הסוגריים המסולסלים הוא התוכן של הפונקציה, והוא הקוד שירוצץ כאשר הפונקציה תופעל. תחילה אנחנו צריכים למצוא את האלמנט אותו אנחנו רוצים לשנות

```
<script>
  function changeColor () {
    var el = document.getElementsByClassName('some-text')[0];

  }
</script>
```

טוב, צריך להסביר כאן כמה דברים. אנסה להסביר זאת במילים פשוטות: אנחנו ניגשים לדף ה-html על ידי המילה document, ובו אנחנו מחפשים את כל התגיות שיש להן class השווה ל-some-text. מכיוון שיש יותר מאלמנט אחד שיש לו class כזה, אז אנחנו מקבלים רשימה של אלמנטים מבלי לדעת מראש כמה איברים יש ברשימה הזאת. מכיוון שאנחנו יודעים שבדף יש רק אלמנט אחד שכזה, אנחנו מבקשים את האלמנט הראשון ברשימה הזאת על ידי הסימון [0]. הערה: בשפות תכנות הספירה של בתוך רשימות ומערכים מתחילה בספרה אפס. איבר [0] הוא האיבר הראשון, איבר [1] הוא האיבר השני, [2] הוא האיבר השלישי וכן הלאה.

את האלמנט שקיבלנו אנחנו שומרים בתוך משתנה שנקרא לו el.

יפה, עכשיו שיש לנו את האלמנט אנחנו נרצה לשנות לו את הצבע. אנחנו עושים זאת על ידי החלפת הערך של ה-color שלו באמצעות הפקודה הבאה

```
el.style.color = 'blue';
```

לכל אלמנט בדף יש style אשר מכיל את כל הגדרות העיצוב שלו. אנחנו יכולים לפנות לכל אחת מהגדרות הללו ולשנות אותן.

עכשיו משעשינו את כל העבודה היפה הזאת, אם תשמרו ותרעננו, תראו את הכפתור הנחמד שלנו, אבל כשתלחצו עליו הוא לא יעשה כלום! הסיבה לכך היא שלא ציינו בשום מקום שלחיצה על הכפתור אמורה להפעיל את הפונקציה changeColor. יש כמה דרכים לעשות זאת, הפשוטה יותר היא להגדיר זאת בתוך ה-html, על גבי הכפתור. לכל אלמנט שנמצא בדף יש מגוון אירועים שניתן "להאזין" להם. לדוגמא, במקרה שלנו אנחנו רוצים "להאזין" לאירוע לחיצה על הכפתור. כחלק משפת html ניתן להגדיר האזנה לאירוע מסוים בתוך התגית עצמה. במקרה שלנו האירוע נקרא "click", ואנו מאזינים לו באמצעות הוספת התכונה "onClick" לתגית של הכפתור

```
<button onClick="">Click me!</button>
```

בתוך הגרשיים של האירוע, ניתן להגדיר לדפדפן מה אנו רוצים שיקרה כשהאירוע הזה קורה. במקרה שלנו כאשר אירוע click מתרחש אנחנו רוצים שהפונקציה changeColor תופעל. זה נראה בדיוק כמו שזה נשמע

```
<button onClick="changeColor()">Click me!</button>
```

כעת משחיברנו בין אירוע הלחיצה על הכפתור לבין הפונקציה שלנו, תשמרו ותרעננו ו... הופ! קסם (-):

לפני שנסיים, אני רוצה לקחת את הפונקציה שבנינו ולהוסיף לה טיפת לוגיקה. אני רוצה שאם הטקסט כבר בצבע כחול, אז לחיצה על הכפתור תשנה את הטקסט לירוק. ובכל מצב אחר שתשנה אותו לכחול.

למעשה הגדרנו כאן תנאי שנראה כך  
**אם** צבע הטקסט הוא כחול  
**אז** תשנה את צבע הטקסט לירוק  
**אחרת** תשנה את צבע הטקסט לכחול

או בשפת JS

```
If (el.style.color === 'blue') {  
    el.style.color = 'green';  
} else {  
    el.style.color = 'blue';  
}
```

ודאי שמתם לב לשלושת סימני השווה (===) אשר מופיעים בשורה הראשונה. ב-JS זוהי פעולת בדיקה אשר מחזירה תשובה של נכון או לא נכון. או ב-JS היא מחזירה true או false.

**אם** התנאי הוא true  
**אז** בצע את הפקודה הבאה (כל מה שבין הסוגריים המסולסלים שאחרי ה-if)  
**אחרת** בצע את הפקודה הבאה (כל מה שבין הסוגריים המסולסלים שאחרי ה-else)

כעת ה-script שלנו יראה כך

```
<script>
  function changeColor () {
    var el = document.getElementsByClassName('some-text')[0];

    If (el.style.color === 'blue') {
      el.style.color = 'green';
    } else {
      el.style.color = 'blue';
    }
  }
</script>
```

נשמור ונרענן. כעת כל לחיצה על הכפתור תשנה את צבע הטקסט.

### סיכום

אני מקווה שהמטרה שלנו הושגה באמצעות המיני-מדריך הזה, והיא שתבינו קצת יותר טוב מה המשמעות הבסיסית של להיות מתכנת front-end. לא, אתם לא יכולים לקרוא לעצמכם מתכנתים כרגע, הדרך ארוכה עד שתגיעו לרמה המינימלית כדי למצוא עבודה ראשונה בתחום. אבל לפחות תוכלו לקבל החלטה קצת יותר מושכלת באשר להאם זה תחום שיעניין אתכם, האם תהנו מתהליך הלימוד עצמו כמו גם מהעבודה. אחרי הכל, ואני מציין זאת שוב ושוב בבלוג שלי, מתכנתים לעולם לא מפסיקים ללמוד לעולם.

כרגיל, אתם מוזמנים לפנות אליי בכל שאלה, הערה, הצעה לתיקון במדריך, שאלות מקצועיות.... בקיצור תרגישו בבית :-)

אסף דוד

[asafusan@gmail.com](mailto:asafusan@gmail.com)

### אז מה עכשיו?

במאמר "[איך ללמוד תכנות](#)" אני מפרט את התובנות שלי מתהליך ההכשרה העצמית שעברתי, בסינון מה שלא עבד לי כמובן, ומנסה לבנות את הדרך הנכונה ביותר וגם המהירה ביותר להגיע לרמה בה תוכלו למצוא עבודה כמתכנתים.

אני רק יכול לשתף אתכם, שבדיעבד ההחלטה שלי ללמוד תכנות הייתה אחת ההחלטות הטובות והחשובות בחיי.  
תהנו!