

# על ג'אווהסקריפט

ג'אווהסקריפט החלה את צעדיה הראשונים ב-1993 כשפה שפועלת בסביבת דפדפן ונועדה להעשיר דפי HTML. עם ג'אווהסקריפט יכולנו ליצור אנימציות ופידבק למשתמשים – כל דבר שהיה קשור לאתרי אינטרנט דינמיים, למשל דברים שנראים לנו היום מאוד טריוויאליים ופשוטים כמו לחיצה על כפתור שעושה פעולה כלשהי בדף. למרות שההתחלה שלה הייתה צנועה, ברנדון איאך, ממציא השפה, יצר אותה מלכתחילה כשפה גמישה מאוד. הגמישות הזו, וגם חוסר ההבנה של רבים מהמתכנתים שהשתמשו בה בנוגע לעקרונות הבסיסיים שלה, גרמו ללא מעט מתכנתים בשפות אחרות לזלזל בה. גם השם שלה לא סייע לתדמית. השם ג'אווהסקריפט נקבע מסיבות שיווקיות בלבד – JAVA היא שפת תכנות פופולרית, ואנשי נטסקייפ חשבו שזה יוסיף לתדמיתה. בפועל השם הזה לא ממש עזר, ואין כמובן שום קשר בין ג'אווה לג'אווהסקריפט.

על אף ההתחלה הקשה, ג'אווהסקריפט הפכה לפופולרית מאוד. בשנת 1996, חברת נטסקייפ העבירה את השליטה על הסטנדרטים של השפה אל ארגון ECMA, ארגון אירופי (היום בין-לאומי) המתמחה בתקינה. המהלך הוביל לשחרור הספציפיקציה של השפה, שידוע בשם ECMAScript, וג'אווהסקריפט "התיישרה" לפי התקינה של ECMAScript. משנת 1997, שנת שחרור ECMAScript, ג'אווהסקריפט, כפי שהיא מיושמת בדפדפנים שונים, עוקבת אחר התקינה של ECMAScript, שהיא בעצם "תוכנית המתאר", וג'אווהסקריפט עצמה היא היישום. לכל גרסה יש מספר משלה בצמוד למילים ES (ראשי תיבות של ECMAScript).

מיקרוסופט התנגדה בתחילה ליישום השפה ויישמה שפה משלה בשם Jscript בדפדפן אינטרנט אקספלורר, שהייתה בנויה בדומה לג'אווהסקריפט. למרות שהייתה יריבות גדולה בין אנשי מיקרוסופט לאנשי ECMA, שנוצרה כתוצאה מפיתוח שתי שפות שנשענות על שני תקנים מתחרים, העקרונות של ג'אווהסקריפט שולבו גם בגרסה של מיקרוסופט. הפופולריות של השפה עלתה כאשר מקרומדיה (יוצרת פלאש) שיתפה פעולה עם ארגון ECMA ושילבה את עקרונות השפה בשפת Actionscript, ששימשה את תוכנת פלאש שהייתה פופולרית מאוד אז.

בשנת 2008 נפגשו אנשי מיקרוסופט ו-ECMA באוסלו והחלו בשיחות שלום. בניגוד לשיחות שלום אחרות שהתקיימו באוסלו, שיחות השלום האלו הסתיימו בהצלחה. תקן ES5, הגרסה הרביעית של ג'אווהסקריפט, שוחרר ויושם בכל הדפדפנים שהיו קיימים אז.

מאז, התפתחות השפה והתפוצה שלה הואצו דרמטית. דפדפן כרום, שמריץ ג'אווהסקריפט באופן יוצא דופן, נכנס אל השוק בסערה ואפשר למפתחי ג'אווהסקריפט לכתוב סקריפטים שפועלים על מנוע V8 העוצמתי של כרום ולהריץ ג'אווהסקריפט במהירות מסחררת. השימוש ב-AJAX – תקשורת אסינכרונית עם השרת – נכנס לפעולה, החליף שיטות מיושנות כגון Long polling ואפשר לאתרים לספק חוויות שימושיות מדהימות למשתמשים עבור המשתמשים. בשנים האחרונות, פריימוורקים וספריות ג'אווהסקריפט אפשרו פונקציונליות מורכבת מאוד וספריות אחרות אפשרו כתיבה של ג'אווהסקריפט גם לטלפונים ניידים ואפילו בקלות. הראשונות שבספריות האלו נקראו MooTools ו-

jQuery והן אפשרו לכל מתכנת לכתוב אפליקציות בקלות. הספריות האחרונות נקראות ריאקט, אנגולר ו-vue והן מאפשרות לבנות תוכנות מורכבות מאוד על גבי הדפדפן (צד הלקוח). ג'אווהסקריפט לא נותר מוגבל רק לצד הלקוח, כלומר לדפדפנים ולמכשירי קצה אחרים; המימוש של ג'אווהסקריפט לצד השרת, הידוע בכינוי node.js, הפך לפופולרי גם בשרתים. ג'אווהסקריפט מריצה כיום אפליקציות מורכבות גם בצד השרת, במיוחד אפליקציות שצריכות לבצע קריאות ולשרת מיליוני משתמשים. כיום אפשר למצוא ג'אווהסקריפט בכל מקום: באתרי אינטרנט, באפליקציות של טלפונים ניידים, באפליקציות המיועדות למחשבים רגילים וכמובן בשרתים. הביקוש למתכנתי ג'אווהסקריפט נמצא בשיאו ואין זה פלא – אפשר לעשות עם השפה הזו המון דברים יישומיים כמעט מאפס. יש כל כך הרבה ספריות וכלי עזר, עד כדי כך שכמעט בכל שבוע יוצאת ספרייה שימושית חדשה. בעזרת ידע מועט אפשר לעשות הרבה מאוד. מה שחשוב הוא ידע בסיסי בשפה.

בשנים האחרונות, תקן ES מתעדכן בכל שנה ומתווספים אליו תכונות ושימושים חדשים. ספר זה מעודכן לגרסה האחרונה של ECMAScript. חשוב לזכור שאם התקן מתעדכן, אין פירוש הדבר שהעדכון החדש מופיע מייד בדפדפנים שמריצים ג'אווהסקריפט או בשרתים שמריצים ג'אווהסקריפט, ושלווקח זמן עד שהעדכונים החדשים ביותר עושים את דרכם אל הדפדפנים/שרתים שכולנו משתמשים בהם. אם שמעתם מפתחי אינטרנט "מקטרים" על דפדפנים ישנים – זו בדיוק הסיבה.

זה המניע לכתיבת הספר. הבן שלי, כיום מתכנת בזכות עצמו, ניסה ללמוד ג'אווהסקריפט מאפס ולא הצליח למצוא ספרים בעברית. החומר שיש כיום בעברית בנוגע לג'אווהסקריפט הוא דל ומיושן. חלק מחוברות העזר הנמצאות בבתי הספר מתייחסות לתקנים שהפסיקו להיות בשימוש בשנת 2007! התייחסות אמיתית לתקנים החדשים ביותר של השפה שיצאו בשנת 2017 אין בנמצא בעברית. התחלתי לכתוב הסברים עבור בני ומפה לשם הבנתי שאני חייב לקחת את זה הלאה.

ג'אווהסקריפט היא שפה שקל ללמוד אותה. בניגוד לשפות אחרות, לא נדרשת סביבת שרת מורכבת או כלי פיתוח שעולים כסף. לא נדרש ידע מקיף במדעי המחשב. כל שצריך הוא לפתוח Notepad במחשב, לפתוח דפדפן ולהתחיל ללמוד ולפתח. צריך גם הדרכה נכונה ומשמעת עצמית. אני מקווה שבספר הזה תמצאו לפחות הדרכה נכונה. המשמעת העצמית – עליכם. אני מאמין שכל שתתקדמו בספר תראו את פירות הלימודים, הניצוץ בעיניים יידלק יותר ויותר ולא תצטרכו עוד משמעת עצמית – אתם פשוט תתאהבו ב-JS בכלל ובפיתוח לוו בפרט. אל תדלגו על הפרק שבו אני מסביר איך ללמוד; זהו הפרק החשוב ביותר בספר.

אני מאחל לכם הצלחה רבה, בין אם אתם מתכנתים בתחילת דרככם או בין אם אתם מתכנתים ותיקים שמשמשים בספר כדי לחזק את הידע שלכם.

# איך לומדים

לא למדתי מדעי המחשב באוניברסיטה או במכללה. למען האמת, עד גיל מאוחר מאוד לא למדתי תכנות בעזרת מדריך. רוב מה שאני יודע למדתי ללא הדרכה, ומכמה סיבות: הראשונה היא שכאשר התחלתי את צעדיי הראשונים בתכנות, בשנת 1996, החומר שהיה זמין באינטרנט היה דל מאוד. על חומר בעברית לא היה מה לדבר, והחומר באנגלית בדרך כלל סיפק רק את הדוקומנטציה. אני לא ממש מתגאה בכך; למידה לבד ללא הדרכה היא למידה מאוד לא יעילה. הדרכה משמעה לא רק מרצה מנוסה, היא גם אתר אינטרנט שבו יש הסבר מקיף, פורום או קבוצה ברשת חברתית זו או אחרת (לא רק פייסבוק) שיש בהם אנשים מנוסים שיכולים לסייע או להפנות לחומרים לחומרי עזר. היא גם מקום שבו אפשר לתרגל ולהתנסות. למרבה המזל, בימים אלו קיימים שלל חומרים, עזרה וסיוע. גם הספר הזה הוא הדרכה. לא תידרשו לצלול לתוך הדוקומנטציה של ECMAScript על מנת להבין את השפה, אבל גם בעזרת הספר תימצא דרך טובה ללמידה. כיוון שרוב הזמן למדתי ללא הדרכה, גיבשתי כמה עקרונות ללמידה שהכנסתי אותם לספר הזה. אני ממליץ לכם לעקוב אחריהם.

## ארגנו לעצמכם סביבת עבודה מסודרת

הפרק הראשון עוסק בבניית סביבת העבודה והוא הפרק החשוב ביותר. ארגנו את המחשב שלכם, סדרו לעצמכם תיקייה מאורגנת שבה תשמרו את כל התרגולים. אם המחשב שלכם מקפיץ התראות של עדכוני ג'אווה או משהו בסגנון דומה, טפלו בכך. ודאו שאתם יכולים להיכנס לאתר הלימודי שמלווה את הספר ושהסיסמה שלכם תקינה ופועלת.

## קראו את הפרקים לפי הסדר

הפרקים לא פוזרו באקראי אלא תוכנו בסדר מסוים. אין טעם ללמוד AJAX לפני שמבינים איך תכנות אסינכרוני עובד. אין טעם ללמוד תכנות אסינכרוני לפני שמבינים איך קולבקים עובדים. ואם גם זה נשמע לכם ג'יבריש, זה סימן שצריך להתחיל מההתחלה. קראו כל פרק לפי הסדר.

## קראו כל פרק פעמיים וכתבו לעצמכם את כל הדוגמאות

קראו את הפרק פעם אחת קריאה שוטפת. לאחר מכן קראו אותו שוב. הפעם קחו את כל דוגמאות הקוד, העתיקו אותן לסביבת העבודה שלכם ושחקו איתן! נסו לשנות את הערכים, לגרום לשגיאות, לכתוב קוד דומה.

אני מאמין שקוד לומדים דרך הידיים ולא רק דרך הראש. חשוב להבין את התיאוריה והרעיונות מאחורי מה שאנחנו מנסים לעשות, אך ללא מימוש, הידע הזה יתנוון וייעלם, בדיוק כמו בשפת דיבור. אם לא נשתמש ונתרגל, גם הלימוד התיאורטי המעמיק ביותר לא יהיה שווה הרבה. הקריאה הראשונה נועדה לתיאוריה, הקריאה השנייה נועדה לתרגול. לימוד שפה הוא לא מרוץ! קחו את הזמן, כתבו את כל דוגמאות הקוד ונסו לכתוב כאלו משלכם.

## פתרו את התרגילים לדוגמה בסוף כל פרק

בסוף כל פרק יש תרגילים לדוגמה. נסו לפתור אותם. אל תתייאשו מהר ואל תרצו אל הפתרון אלא שברו קצת את הראש. הצלחתם לפתור? נהדר. קראו את הפתרון המוצע ואת ההסבר וראו אם הם דומים לשלכם או שונים במקצת. יש יותר מפתרון אפשרי אחד לכל בעיה...

## התייעצו עם אחרים

יש לא מעט קבוצות בעברית (בפייסבוק אבל לא רק) המיועדות ללימוד ג'אווהסקריפט ולדיון בה. מצאו את זו שהכי נוח לכם בה. אל תהססו לשאול שם שאלות. לא הבנתם משהו? דוגמה כלשהי לא הייתה מובנת? הפתרון לתרגיל לא היה ברור מספיק? שאלו שם, ובעברית. אל תהססו לקרוא ולהשתתף בדיונים, בין אם זה לסייע למישהו שהידע שלו דל יותר משלכם או לקרוא דיון על מהות השפה. אל תשכחו להבין את רוח הדברים בקבוצה ולא להציק או להעיק. רוב מי ששתתף בקבוצה הם אנשים עובדים שלא דווקא זמינים להודעות מיידיות.

## הגיעו לסדנאות ולמיטאפים

לא מעט חברות מארגנות בחינם סדנאות, מיטאפים ומפגשים שבהם מתכנתים מציגים את ג'אווהסקריפט ומרצים עליה. כדאי מאוד להגיע למפגשים האלו, לא רק על מנת לשמוע את התכנים ולהשתתף בסדנאות אלא גם להכיר אנשים אחרים בתעשייה. עם חלקם אתם תכתבו בקבוצות הפייסבוק ל-JS. קהילת מפתחי ה-JS בארץ היא קהילה מאוד שיתופית וקרובה, ורבים בה מכירים זה את זה.

הספר לא שווה הרבה בלי תרגול מקיף ושימוש בו. אל תעברו לפרק הבא לפני שתסיימו את כל התרגולים שקשורים לפרק שקראתם. זה לא קריטי, זה סופר-קריטי.

# התקנת סביבת עבודה ודרך הלימוד

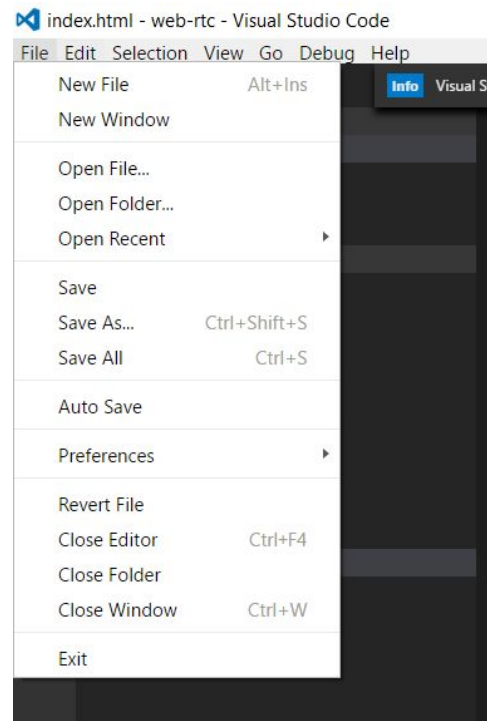
ג'אווהסקריפט יכולה לרוץ בסביבת שרת או דרך הדפדפן. איך זה בדיוק עובד? ג'אווהסקריפט נמצא על קובץ טקסט. כן, בדיוק כמו זה שאפשר ליצור עם כתבן הטקסט (Notepad) שיש בכל מערכת חלונות. בדפדפן מותקן כלי שלוקח את קובץ הטקסט הזה ומריץ אותו ואת הפקודות שנמצאות בתוכו. אם הדפדפן היה אדם, הוא היה פותח את קובץ הטקסט וקורא את מה שיש בתוכו, למשל: "לך ימינה ופתח את הדלת", ועושה בדיוק מה שכתוב. הפעולה הזו נקראת בלשון הפופולרית "רינדור", מלשון render בלעז. הדפדפן לוקח את קובץ הג'אווהסקריפט ומריץ אותו. קובץ הג'אווהסקריפט יכול לעשות כל מיני דברים ולהציג אותם או לא להציג אותם.

איך הדפדפן טוען את קובץ הג'אווהסקריפט? יש כמה דרכים לעשות זאת, אבל כרגע אנחנו רוצים ללמוד איך לכתוב ג'אווהסקריפט ולראות אותו פועל על מנת להבין את כללי השפה ולכתוב משהו באופן ראשוני ביותר. החלק החשוב והקשה ביותר שעומד לפנינו הוא יצירת סביבת העבודה, כלומר סביבה ממוחשבת שבה אפשר להקליד ג'אווהסקריפט ולראות אותו עובד. סביבה זו היא חשובה מאוד כאשר אנו לומדים, כיוון שלימוד של שפת תכנות נעשה ראשית כול "דרך הידיים" וחשוב מאוד לא רק לקרוא אלא גם לתרגל. וכדי לתרגל אנחנו צריכים סביבה שמאפשרת להקליד פקודות שפה, לשמור ולראות את הפלט. לדעתי, התקנת הסביבה היא החלק הקשה ביותר בתחילת לימוד שפה חדשה וגם החשוב ביותר. לפיכך כדאי להיאזר בסבלנות, לקחת נשימה ארוכה ולזכור שדווקא עכשיו החלק הקשה ביותר.

בואו נתחיל עם עורך טקסט טוב. למרות שאפשר להשתמש בנוטפד, הוא לא מציע צביעת קוד לצורך עזרה בקריאות וכמו כן לא מציע יצירת הזחות בקלות. יש כמה עורכי טקסט המותאמים במיוחד לכתובת ג'אווהסקריפט. אציין כמה מהם. בחרו באחד! רובם זהים למדי ומכילות יכולת עריכה בסיסית של HTML, CSS וג'אווהסקריפט.

**שימו לב:** מתכנתים מנוסים יותר לא משתמשים בעורך הטקסט הפשוט יותר אלא בעורך טקסט משוכלל יותר שנקרא IDE או Integrated Development Environment – סביבת פיתוח משולבת. הסביבה הזו מאפשרת השלמת קוד, מצינת כתבים שגיאות בכתיבה וגם יכולה להריץ את הקוד עצמו.

IDE מעולה הוא Visual Studio Code. הוא חינמי, מבוסס קוד פתוח, כתוב בג'אווהסקריפט (כך, כן) נתמך על ידי מיקרוסופט וניתן להורדה פה: <https://code.visualstudio.com>. אחרי ההתקנה תוכלו לפתוח את התוכנה, לבחור ב-File ואז לפתוח את התיקיה שבחרתם ולפתוח או ליצור בה קבצים.



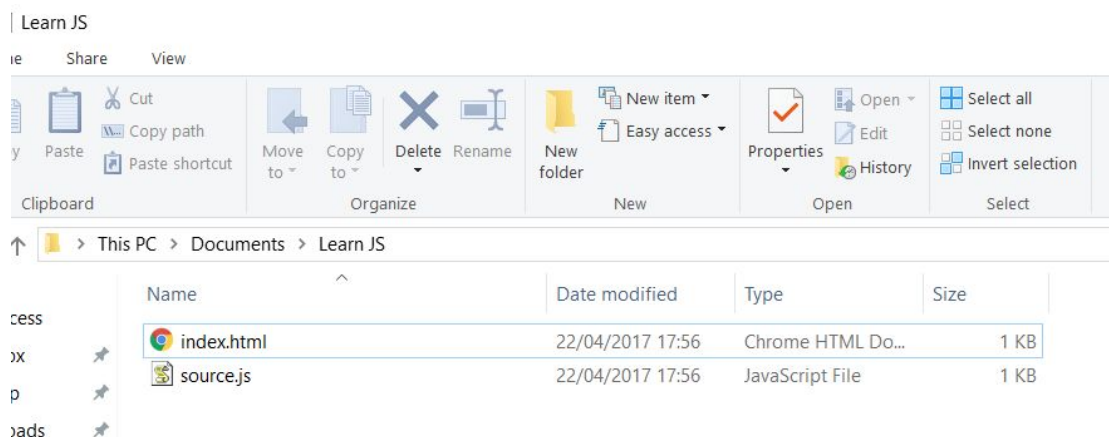
IDE מוצלח אחר הוא Atom. גם הוא חינמי, מבוסס קוד פתוח וגם הוא... כתוב בג'אווהסקריפט. הוא נתמך על ידי גיטהאב. הוא ניתן להורדה פה: <https://atom.io> הוא די דומה ל-Visual Studio code ואחרי ההורדה וההתקנה שלו אפשר להפעיל אותו בקלות.

עורך הטקסט נוסף שנחשב לאמין וטוב הוא תוכנה חינמית בקוד פתוח שנקראת Notepad+++. הוא ניתן להורדה בקישור הבא: <https://notepad-plus-plus.org/download> והוא בסיסי יותר משני קודמיו.

שימוש ב-Visual studio code או ב-Atom הוא מומלץ יותר כיוון שיש בו "השלמה אוטומטית" של פקודות נפוצות בג'אווהסקריפט, דבר המקל מאוד את הלמידה. כמו כן, הוא מציג שגיאות בקוד כבר במהלך הכתיבה, עוד לפני ההרצה.

הסביבה הטובה ביותר ללימוד היא יצירת קובץ HTML שטוען קובץ ג'אווהסקריפט. קובץ HTML הוא קובץ שהדפדפן יודע לפרש ולהציג והוא יכול לקרוא לקובץ ג'אווהסקריפט באופן כזה שהדפדפן ירנדר

אותו. כאמור, רינדור הוא הרצת הפקודות שנכתוב בקוד הג'אווהסקריפט. יש ליצור במחשב תיקייה – זה יכול להיות ב"המסמכים שלי" או על שולחן העבודה – ובתוכה ליצור קובץ ששמו source.js וקובץ ששמו index.html.



**ודאו** שמערכת החלונות או המק שלכם תומכת בתצוגת שם הקובץ המלא, כולל הסימנת (Extension). אחרת, קובץ ה-index.html יהיה בעצם index.html.txt.

בקובץ ה-HTML כתבו את הקוד הבא:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body></body>
  <script src="./source.js"></script>
</html>
```

בתוך קובץ ה-source.js כתבו את הטקסט הבא:

```
document.write('Hello World!');
```

אחרי ששמרתם את תוכן שני הקבצים, פתחו את הקובץ בדפדפן כרום או בפייירפוקס (לא באדג'). אם הכול תקין, תראו שכתוב על המסך: *Hello World!* כתבתם את הג'אווהסקריפט הראשון שלכם!

**שימו לב:** זה השלב המועד ביותר לפורענות, שעלול לתסכל מאוד, אבל הוא שלב חשוב מאוד ואסור להרים ידיים ולהתייאש. אם פתחתם את הקובץ ודבר לא הופיע, נסו את הדברים הבאים:

1. בדקו שאכן קראתם לקבצים index.html ו-source.js. הבדיקה צריכה להתבצע באמצעות הכפתור הימני של העכבר בחלונות, כדי למנוע מצב שבו מערכת ההפעלה הוסיפה תוספות לשמות ו-index.html נשמרה בשם index.html.txt.
2. בדקו שאתם פותחים את הקבצים בדפדפן כרום או פיירפוקס. בדקו שאין תוספים מיוחדים לדפדפנים שעלולים לחסום את הרצת הקובץ על ידי הרצה של מצב פרטיות.
3. בדקו שהקלדתם את הטקסט כשורה בקובץ source.js ללא רווחים או תווים מיוחדים.
4. נסו להשתמש ב-Atom או ב-Visual studio code. שימו לב שאין התראות על שגיאות הקלדה. כאן למשל מובאת דוגמה של שגיאת הקלדה שביצעת. אם קיבלתם התראה כזו, בדקו שוב שלא טעיתם בגרשיים או הכנסתם רווחים מיותרים:

```

1 document.write('Hello World!');

```

אם אתם משתמשים ב-Visual studio code או Atom, סביבת כתיבת הקוד שלכם אמורה להיראות כך:

```

1 <!doctype html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5   </head>
6   <body>
7 </body>
8   <script src="./source.js"></script>
9 </html>
10

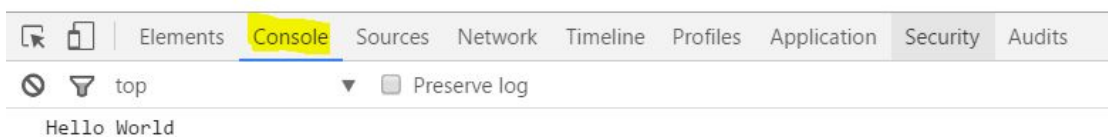
```

מצד שמאל תוכלו לראות את כל הקבצים בתיקייה. כדאי ליצור תיקייה מיוחדת בשם learn JS או בסגנון בשם דומה ולא לשים את כל הקבצים בתיקייה משותפת כמו "המסמכים שלי". מצד ימין תוכלו לצפות בתוכן הקבצים. בצילום המסך רואים את תוכן הקובץ index.html. אם תקלידו דבר מה, תוכלו לראות שיש השלמה אוטומטית של קוד HTML ואם תקלידו בקובץ source.js שמכיל



את הג'אווהסקריפט תראו שיש השלמה אוטומטית של פקודות ג'אווהסקריפט. נוסף על כך, תקבלו גם התראה על קוד לא תקין.

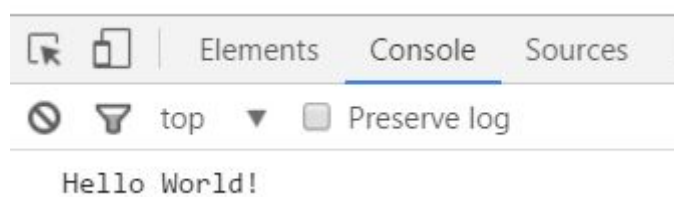
כיוון שאתם כבר מפתחי ג'אווהסקריפט מקצועיים, תלמדו לעבוד עם הקונסולה של הדפדפן. מדובר בממשק מיוחד שמאפשר "לדבג" את ג'אווהסקריפט. הפועל "לדבג" כוונתו להסתכל על צפונות הרינדור ולראות ממש את הפלט של השפה. נשמע מסובך? יש להבין איך הדפדפן מנסה להריץ את הקוד שלו (מה שנקרא "רינדור", מלשון הרצה, באנגלית), ולקבל מידע נוסף במידה שהוא נכשל, כלומר "זורק" שגיאה צבועה באדום בקונסול ואף מפנה לשורה הבעייתית. פתחו את כלי המפתחים של הדפדפן. בכרום ובפייירפוקס לחצו + i Ctrl + Shift אם יש לכם חלונות או Cmd + Shift + i אם יש לכם מק. אפשר לעשות את זה גם דרך התפריט העליון בשני הדפדפנים. לאחר פתיחת כלי המפתחים, לוחצים על Console.



בואו נבדוק מה אפשר לעשות עם הקונסולה. היכנסו אל source.js והחליפו את הטקסט ל:

```
;!console.log('Hello World');
```

טענו מחדש את הדף באמצעות Ctrl + F 5 בחלונות או Cmd + R במק. הטעינה מחדש חשובה. הדפדפן לא יודע שהוכנסו שינויים בקובץ הג'אווהסקריפט, ויש לגרום לו לטעון מחדש את קובץ הג'אווהסקריפט על מנת להריץ את הפקודות החדשות. אחרי הטעינה מחדש הסתכלו על לשונית ה-Console. אם הכול תקין, המסך יהיה ריק, אך בקונסולה תראו *Hello World!* יש!



**חשוב:** אל תדלגו על השלב הזה. בכל שלבי הלימוד כדאי להשתמש בקונסולה, שהיא הרבה יותר נוחה להצגה. במידה שמהו לא עובד, אנא בדקו ש:

1. האם השלמתם את השלב הקוד? הצלחתם להציג Hello World על המסך?
2. האם שמרתם את הקובץ לאחר השינויים?
3. האם טענתם מחדש באמצעות Ctrl + F 5 את הדפדפן?

מדובר בסביבת העבודה הטובה ביותר ללימוד ג'אווהסקריפט, אך יש סביבות עבודה נוספות. ברשת יש אתרים המאפשרים לכתוב ג'אווהסקריפט ישירות ולהריץ את הקוד דרכם ולראות את התוצאות. אתר מפורסם וחשוב כזה נקרא <http://codepen.io> ואפשר להקליד בו פקודות של ג'אווהסקריפט. צרו שם חשבון וצרו "pen" חדש. בדקו שיש אפשרות להכניס קוד HTML, CSS ו-JS שהוא בעצם קיצור של ג'אווהסקריפט. אפשר להכניס את הקוד ולראות את התוצאות על גבי דף מדומה או על גבי הקונסולה של הדפדפן.

מכאן, דרך הלימוד תהיה פשוטה למדי. אני אסביר על תכונות מסוימות של השפה ואתן דוגמאות. מומלץ בחום רב להעתיק את הדוגמאות אל קובץ ה-source.js ולהריץ את הג'אווהסקריפט כדי לראות איך זה עובד באמת.

בסוף כל פרק יש תרגילים לתרגול עצמאי ומומלץ בחום רב לנסות אותם על סביבת העבודה. אי אפשר ללמוד שפה על ידי קריאה תיאורטית בלבד וכדאי ורצוי לתרגל, לתרגל, לתרגל. את התרגול עושים רק בסביבת עבודה יציבה. לפיכך, אנא אל תדלגו אל הפרק הבא לפני שיש לכם סביבת עבודה יציבה. מומלץ מאוד לעבוד ב-Visual studio code החינמית והמומלצת.

תרגול לדוגמה:

במקום *Hello World* גרמו לקונסולה להדפיס את המילים *Ahla Bahla*.

פתרון:

היכנסו לקובץ source.js, מחקו את הטקסט שיש שם והדביקו במקומו את:

```
console.log('Ahla Bahla');
```

שמרו את הקובץ, פתחו את הקובץ index.html שטוען את הקובץ source.js ורעננו את הדף. לחצו על `Ctrl + Shift + i`, לחצו על הלשונית Console וראו את התוצאות.

תרגול לדוגמה:

גרמו לקובץ ה-html לטעון קובץ JS ששמו הוא targil.js ושמדפיס לקונסולה: *I am new file*.

פתרון:

צרו קובץ targil.js באותה תיקייה של הקובץ index.html. יש לוודא שזה שמו האמיתי של הקובץ ושמערכת ההפעלה לא מצמידה לקובץ סיומת txt. את זה עושים על ידי בדיקה בהגדרות התצוגה של מערכת ההפעלה. פתחו את קובץ ה-HTML בעזרת עורך טקסט, מומלץ Visual studio code, ושנו את שם קובץ הג'אווהסקריפט ל-targil.js.

```
<!doctype html>
<html>
  <head>
```

```
<meta charset="utf-8">
</head>
<body></body>
<script src="./targil.js"></script>
</html>
```

בקובץ targil.js כתבו את השורה הזו:

```
console.log('I am new file');
```

פתחו את הקובץ index.html בדפדפן ולחצו `Ctrl + Shift + i` על מנת להציג את כלי המפתחים.  
בחרו את הלשונית Console ובחנו את התוצאה.