מדריך Java Script

תוכן עניינים

מבוא

לאחר שהבנו כי יש לכל אתר אינטרנט את שפת הhtml והשלד שלו והמשכנו להבין כי יש עיצובים שונים לכל דף בצורת הcss, כעת נוכל גם ליצור ממשקים ולכתוב סקריפטים אשר יהפכו את האתר לדינאמי ובעל שינויים תוך כדי תנועה. למשל, המשתמש יוכל להזין את שם המשתמש שלו והדפדפן יזכור...או למשל נוכל לבנות מערכת ציונים ולהציג ממוצע... אלו הם פעולות לוגיות אשר לא ניתן לבצע על ידי html או css, אלא דרך java script.

גם כאן ניתן לכתוב בnotepad++ תחת 'שפה' ולבחור בjava script או ליצור תגית script בתור הhead שלנו ובה לחולל את כלל הסקריפטים. מומלץ לקרוא את פרק הלוגיקה לפני התחלת הכתיבה בjava script.

במדריך זה אסביר ואפרט על יסודות השפה, כיצד לכתוב בה וכיצד ניתן לבנות ממשקים וליצור פעולות שונות באמצעות הjava script. רצוי מאוד גם לעשות חזרה על שיעור הform המפורט משלב הhtml. צפו בסרטון המצורף והתחילו להיכנס אל תוך עולם הjava script...כתיבת הקודים של אתרי האינטרנט.

יסודות השפה

בשיעור זה, אלמד אתכם איפה כותבים סקריפטי Js בכלל וכן גם אדבר על משתנים (אם כי, רצוי ואף מחוייב ללכת למדריך 'מבוא לתכנות' על מנת להכיר מושגים כמו משתנה, תנאי, לולאה, מערך ועוד).

אז איפה כותבים JS?

- כתיבה בתוך תגית HTML

ניתן לכתוב בתוך התגית עצמה פקודה כלשהי (כמו למשל הקפצת הודעה, עליה נלמד בשיעור הבא). מה שצריך לכתוב בתוך התגית זו תכונה של onclick ולאחריה לרשום = ובגרשיים את הפקודה שנילמד בהמשך. זו הדרך ליצור סקריפט בתוך התגית. בדומה לCSS יש אפשרות גם ליצור סקריפט חיצוני. (הקבילו את הonclick לstyle בתוך תגית, כך יהיה לכם הכי קל להבין).

```
<p onclick='???'>
```

- כתיבה בתגית script בhead או בbody

ניתן לכתוב בתוך הhead תגית `<script></script>` ובתוכה לתת את כל הפקודות וההנחיות. (נתחיל עם הפקודה של הקפצת הודעה, כבר בשיעור הבא ותנסו לתרגל בעצמיכם להקפיץ הודעה בכל השיטות הנלמדות בשיעור זה).

בנוסף, ניתן גם לרשום את תגית הscript ובתוכה שלל הפקודות גם בbody. בניגוד לstyle בcss שאותה תמיד נכתוב בhead לפני שכל קטעי הHTML שבbody נוצרו, לעיתים בjs

נצטרך לכתוב את הפקודות דווקא אחרי וזו מהסיבה שתמיד נרצה להעניק פקודה או לדבר עם אלמנט מסוים שהוא כבר קיים בדף ולא לפני שהוא נוצר... (עלול לקרות מצב כזה בו תתנו הנחיה לכפתור מסוים להקפיץ הודעה, אבל הוא כלל לא קיים והמערכת תיקרוס). צריך לשים לב לכך. לעיתים נעדיף שהתגית תהיה דווקא בBody ולמטה.

- קישור לקובץ script חיצוני

גם כאן, בדומה לcss - ניתן לכתוב בתוך קובץ נפרד את כלל הפקודות שלנו. כמובן לתת לקובץ הנ'ל סיומת js בסביבת הפיתוח שלכם ואז לאחר מכן, גשו לhtml שלכם אל הhead או הbody והעניקו רפרנס (קישור) מתאים. איך כותבים זאת? כך –

```
<script type='text/javascript' src='myScript.js'></script>
```

לאחר שהבנו כיצד לקרוא לקטעי הJS, כעת אסביר על משתנים בשפת ג'אווה סקריפט: בניגוד לשפות פיתוח אחרות, בג'אווה סקריפט יש רק משתנה אחת כללי לכל מה שתתנו לו. בין אם תרצו שהמשתנה יכיל טקסט, תמונה, משתנה בוליאני, מספר גדול, מספר קטן וכו'.

החוקיות במשתנים היא ברורה: תמיד כאשר ניצור את המשתנה בפעם הראשונה נרשום var בתחילתו.

לדוגמא – `var tal = 1;`

וכאשר נרצה לבצע עליו מניפולציה כלשהי, כבר המחשב יכיר את המשתנה ולכן אין צורך לרשום לו שוב var.

לדוגמא – `tal * tal;`

הנה קוד שלם אשר יוצר משתנים ומבצע עליהם מניפולציה. האם אתם יודעים מה עתיד להיות הפלט?

```
var num1 = 5;
var num2 = 5;
var sum = num1+num2;
var avg = sum / 2;
```

כל מה שנותר כעת לעשות זה להציג את הפלט של התרגיל. הצטרפו גם לשיעור הבא במדריך העוסק בהקפצה של הודעה וכך נוכל להציג את הפלט הנוצר. השאירו תגובות כאן במדריך או במייל שלי ואעזור בכל שאלה.

Alert

נתחיל עם הסקריפט הפשוט ביותר. בתוך תגית הסקריפט כיתבו את הקוד הבא:

```
alert('hello!');
```

חשוב לציין שגם כאן אחרי כל שורת קוד, בצורה לcss, נצטרך להפריד באמצעות `;` הרובוט שקורא את האתר שלנו יכול להפריד בין קטעים שונים על ידי הנקודה פסיק, שהרי הוא קורא את הכל בצורה של שורה אחר שורה ולא כמונו שמסדרים את האתר לפי רווחים וכדומה...

למעשה, פקודת הalert מציגה לנו פלט כלשהו. כאשר כתבתם hello בתוך גרשיים הדבר גרם לalert להציג את הטקסט hello. אם הייתם שמים רק hello ללא גרשיים, הוא היה בטוח שמדובר בשם של משתנה.

בjs ניתן גם ליצור משתנים כמובן. פשוט לפני כל שם משתנה נוסיף את המילה var ונבצע השמה. שימו לב לדוגמא הבא:

```
var num = 5;
alert(num*num);
```

בקוד הנ'ל גם יצרו משתנה חדש בשם num וגם הצגנו למשתמש את המספר כפול עצמו.
ניתן גם לפרק את num*num למשתנה נוסף ובalert להציג את המשתנה השני. אתם יכולים לשחק עם זה איך וכמה שתרצו כמובן.

הבדל בין + לשאר האופרטורים.
הרובוט יודע להבדיל מתי עליו לבצע פעולה מתמטית בהקשר של כפל, חילוק, חיסור, מודולו... אבל כאשר מדובר בסימן + הוא במצב ברירת המחדל שלו תמיד ילך על חיבור המספרים לידי מחרוזת. שימו לב לכך. לכן, כאשר אנו עושים alert למדיי כדוגמא הבאה, תמיד נשים את פעולת החיבור בתוך סוגריים משלהם.

```
alert('the result is:' + (num+num));
```

כך בjs אנו מבדילים בין חיבור של הוספה לבין חיבור מתמטי.
נסו כעת בעצמיכם, צרו 3 מספרים
והציגו בסוף את השורה הבאה בalert: 'הממוצע של 3 המספרים הוא: X '.

לוגיקה בJs

בשיעור זה, אני מסתמך על כך שכבר עברתם על המדריך של 'מבוא לתכנות'. כעת, אסביר כיצד אופרטורים, תנאים ומושגים בסיסיים שונים בתכנות באים לידי ביטוי בjs.

משתנה

כפי שהסברתי גם בשיעור מס' 2, משתנים באים לידי ביטוי בצורה זהה לחלוטין בין ערכים שונים. אפרט על כך שנית. ראשית, מזכיר כי ניתן לדמות משתנים אל מערכת של ספריות בזיכרון של המחשב. למעשה, משתנים הם כמו מגירות בהם ניתן 'לאחסן' ערכים. על מנת להשתמש במשתנה (לפתוח מגירה ולהכניס לתוכה ערך) עלינו לכתוב את מילת הקריאה var (קיצור של variable) שורת הקוד var x; תיצור משתנה בשם x שכרגע אין בפנוכו שום ערך. משתנה יכול לאחסן בתוכו מספר סוגים של ערכים כגון מספר או תו (אות) או ערך בוליאני (true או false) JS יודעת כבר בעצמה איזה type (סוג משתנה) יושב בכל תא אחסון שכזה. איך היא יודעת? הרי תמיד נרשום var? למעשה, סוג המשתנה נקבע לפי הערך שנשים בתוכו. למשל, כאשר נרשום var x=5; ניצור תא בשם x לאחסון משתנה ונאחסן בתוכו את המספר 5. העובדה שאחסנו בתוכו את המספר 5 קובעת שתא זה מחזיק מספר. וכאשר נרשום var y = '5'; למעשה ניצור תא בשם y לאחסון משתנה ונאחסן בתוכו את התו '5'. העובדה שאחסנו בתוך התא y את '5' קובעת שתא זה מחזיק מחרוזת (תוים). ההבדל בין הפקודות הינו הגרשיים המקיפות את הספרה 5 והיא זו שגורמת ל JS להתייחס אחרת לערך היושב בתא.

מה בנוגע לכל הנושא של 'השמה'? שגם אותו לימדתי בהרחבה בשלב המבוא לתכנות-
שינוי ערך של משתנה תעשה באמצעות האופרטור שווה =
ראשית נגדיר משתנה באמצעות VAR `var name = 'TAL';`
ולאחר מכן נשנה את ערכו `name = 'GAL';`
כבר לא נצטרך לרשום var שהרי התא כבר נוצר. אלא רק לשנות את הערך שבפנוכו.

יתרונות השימוש במשתנים

השימוש במשתנים מאפשר להפריד את הלוגיקה מהנתונים כאשר הפעולות מתבצעות על משתנים ולא על הנתונים עצמם ניתן להשתמש באותו קוד שוב ושוב עם נתונים שונים לדוגמא – אם יש לנו קוד שיודע להמיר טמפרטורה במעלות צלזיוס לפרנהייט והקוד כתוב באמצעות משתנים שאם הערך שונה כל שעלינו לעשות זה לשנות את ערך המשתנה אבל לא לכתוב את הקוד מחדש

שמות משתנים

שם המשתנה נקבע על ידנו, לא ניתן לתת שם משתנה שכולל רווחים נהוג כי כאשר שם משתנה מורכב ממספר מילים אז כל מילה תתחיל באות גדולה כך שניתן יהיה להבין את רצף התווים זוהי שיטה מקובלת ב JS והיא נקראת camel case (כמו דבשות של גמל)

סוגי משתנים שניתן להכניס בתוך הvar שלנו

String – מחרוזת של רצף תווים, מילים או מספרים מוקפים בגרש בודד או גרשיים כפולים
Number - מספרים, כל מספר (דצימלי). גם שלמים וגם עשרוניים (נרשמים ללא גרשים)
Boolean – בוליאני . הערך הוא תמיד true או false (נרשם ללא גרשיים)

אופרטורים מתמטיים

1. חיבור +

2. חיסור –

3. כפל *

4. חילוק /

5. מודול (שארית) %

הוספה בתוך מחרוזת קיימת

ניתן להשתמש באופרטור + למשמעות נוספת מלבד חיבור שני מספרים, בדיוק כפי שנילמד בחלק המבוא לתכנות.

הכוונה היא, שבמידה ויש לנו משתנה שהוא טקסט והjs מזהה שהוא טקסט , אם נשתמש בתוכו בסימן + זה יהיה לחבר בין מחרוזות. כמו בדוגמא הבאה:

```
<script>var name = 'Haitech' + 'World'</script>
```

הערות סינטקס נוספות בכתיבה בjs

פסיק נקודה בסוף כל שורת פקודה!

הערות - שורות שלא יופעלו כשנריץ את הקוד. למעשה, נועד להסביר את הקוד לכשנחזור אליו עוד שנה או כשמתכנת אחר יסתכל בו :

שורה בודדת //

מספר שורות /* */

בנוסף, כל האופרטורים בהקשר של תנאים ולולאות (שווה, לא שווה, גדול מ-, קטן מ-, &&, ||) יהיו בדיוק כפי שנילמד בחלק הלוגיקה כאן באתר. נא לעבור עליו ולדעת אותו על בוריו. בנוסף, גם כל נושא התנאי הפשוט והלולאות אותו הדבר בדיוק. לדוגמא, לולאת for נכתוב כך בג'אווה סקריפט –

```
for (var i=0;i<5;i++);
```

קריאה לאלמנטים מהHTML

ניתן למשוך טקסטים ולדבר עם אלמנטים מתוך הhtml דרך סקריפטים.

עושים זאת באמצעות משתנה אשר קורא לid מסוים בדרך הבאה:

ראשית, מדברים עם הdocument (המסמך עצמו) ואז מאתרים את הid. לאחר מכן נבקש את הhtml שלו –

```
var txt = document.getElementById('tal').innerHTML;
```

כפי שראיתם הכל נעשה על ידי נקודה בין חלק לחלק (בתכנות זה נקרא בין node לnode).

נסו בעצמיכם - צרו תגית p עם id=tal ובפנים טקסט כלשהו. נסו להציג את הטקסט של הp כעת בתוך alert.

בנוסף למשיכה, ניתן גם לשנות. נבצע זאת על ידי השמה.

```
document.getElementById('tal').innerHTML = 'This is A New Text';
```

אותה משיכה כמו טקסטים של div, ניתן למשוך גם את הטקסט שהמשתמש הזין בתוך הinput! הפעם במקום innerHTML נצטרך להחליף לvalue (זאת בהקשר של inputs).

```
var txt = document.getElementById('tal').value;
```

לסיום השיעור - צרו סקריפט אשר מציג בalert את מה שכתוב בתוך הp ובתוך הinput שיצרתם.

פונקציות

בדומה לשיעור 2 ו4, גם כאן אסביר בהרחבה על עולם הפונקציות בתוך הJs.

רצוי לקרוא לפני כן, את נושא הפונקציות במדריך הלוגיקה כאן באתר עולם ההייטק.

כאמור, היופי בפונקציות הוא שאנחנו רושמים את הקוד פעם אחת ואז משתמשים בו שוב ושוב:
מבנה כללי של פוקציה בעולם הjs:

```
function tal() {
alert('hello haitech!');
}
```

המילה function היא מילה שמורה בשפה , `tal` הוא שם הפונקציה שאנחנו קובעים ומיד לאחר שם הפונקציה ישנם סוגרים רגילות (בהמשך נראה מה אפשר לעשות שם) לאחר מכן תוכן הפונקציה עטוף בסוגרים מסולסלות כאשר נרצה לקרוא לפונקציה באמצע קוד אחר, נניח בתוך תנאי כלשהו - הקריאה לפונקציה תראה כך: `tal();` פשוט נרשום את שם הפונקציה, סוגריים ריקות (במקרה זה ריקות) ונקודה פסיק.

למה אנחנו חייבים להזין סוגריים? מתי הן ריקות ומתי מלאות? ראשית, כל קריאה לפונקציה מחייבת פתח סוגרים וסגור סוגריים בצמוד לשם הפונקציה אחרת היא תחזיר את הקוד עצמו.
ובנוסף לכך, שימו לב להגדרת הפונקציה: (היא מכילה סוגריים ריקות) –

```
function tal(num) {
alert(num*num);
}
```

ובעת הקריאה לפונקציה נכתוב: `tal(5);`

כלומר, הנקודה היא פשוטה. אנו יכולים לשלוח מידע בין קריאות שונות לפונקציות שונות!
הכוונה היא, שניתן לשלוח את המספר 5 (או למשל כפי שלמדנו את המספר שהמשתמש הזין בinput) ולבצע עליו מניפולציה בתוך הקוד עצמו.

בנוסף לכל אלה, לא חובה להקפיץ alert או לבצע שינויים בDOM בתוך פונקציות.

ניתן פשוט להחזיר דרכה ערך. הכוונה היא, שכל הפונקציה תהיה משתנה והיא תהיה שווה בסופו של דבר לערך מסוים. מורכב? לא כל כך. זו פשוט עוד דרך לבצע דברים. לשיקולכם –

```
function tal(num) {
return num*num;
}

var num = document.getElementById('input').value;
var sum = tal(num);
alert(sum);
```

לסיום השיעור והנושא עד כה, צרו אתר ובו המשתמש מזין 3 מספרים. הציגו את הממוצע שלהם או בalert או בדרך הHTML, העיקר - דרך פונקציה.

קלט משתמש

בהמשך לשיעור הקודם, הפעם נלמד איך באמצעות לחיצת כפתור אפשר למשוך את טקסט המשתמש הזין לתוך הinput ולבצע עליו את כל המניפולציות והלוגיקות אשר נלמדו במדריך הלוגיקה.

כל שעלינו לעשות זה לבצע את השלבים הבאים:

1. לתת לכפתור בתוך תגית הפתיחה אפשרות של לחיצה ושליחה לפונקציה מוגדרת מראש. לא לשכוח סוגריים עגולים, וזאת מפני שאפשר להזין ערכים בתוך הסוגריים העגולים ולשלוח פרמטרים לפונקציה (ראה ערך שיעורי לוגיקה - פונקציות).

```
<button onclick='tal()'>Click Me!</button>
```

2. בתוך תגית הסקריפט שלנו הפעם נעטוף את הalert שיצרנו עד כה בתוך סוגריים מסולסלים של פונקציה בשם שקבענו בסעיף 1 (tal) בדרך הבאה:

```
<script>
function tal(){
alert('test');
}
</script>
```

עתה, אחרי שהבנו כי אפשר לקרוא לקטעי קוד מסוימים בתזמוני לחיצות על ידי המשתמש - פה אתם נכנסים לתמונה. בהישען על השיעור הקודם, צרו משתנה אשר קולט את הvalue של Input המשתמש. לאחר מכן, הציגו זאת בתוך alert.

עובד לכם? נפלא! כעת תוכלו למשוך את קלטי המשתמש ולבצע עליהם מניפולציות. למשל, קלטו מהמשתמש מספר והציגו את המספר הנבחר כפול 10.

ולפניכם הפיתרון:

```
<div>
   הזן שם: <input type='text' id='txt' onclick='tal()'/>
</div>

<script>
   function tal() {
      var num = document.getElementById('txt').value;
      var sum = num*10;
      alert(sum);
   }
</script>
```

שאלות לתרגול

אנא פיתרו את כלל הכלים הבאים בjs.

תרגישו חופשי לשלוח לי במייל את פתרונותיכם ואעזור למי שמתקשה !

שימו לב, הכלים מכילים גם בעיות לוגיות ונושאים משלב 'מבוא לתכנות'.

תרגיל מס' 1

רמה: בסיסית

צרו משתנה של טקסט. הציגו את הטקסט בalert.

פיתרון:

```
<input type='text' id='txt' />
<button id='show()'>Show It! </button>

<script>
function show(){
  var txt = document.getElementById('txt').value;
  alert(txt);
}
</script>
```

תרגיל מס' 2

רמה: בסיסית

צרו 2 משתנים - שם וגיל. הציגו למשתמש את השורה הבאה: 'שלום, X , גילך הוא Y.'

פיתרון:

```
<input type='text' id='name' />
<input type='number' id='age' />
<button id='show()'>Show It! </button>

<script>
function show(){
  var name = document.getElementById('name').value;
  var age = document.getElementById('age').value;
  alert('שלום, ' + name + ', גילך הוא ' + age);
}
</script>
```

תרגיל מס' 3

רמה: בסיסית

קילטו מהמשתמש 3 מספרים. בנו תוכנה אשר מציגה את הממוצע של שלושתם.

פיתרון:

```
<input type='number' id='num1' />
<input type='number' id='num2' />
<input type='number' id='num2' />
<button id='show()'>Show It! </button>

<script>
function show(){
  var num1 = parseInt(document.getElementById('num1').value);
  var num2 = parseInt(document.getElementById('num2').value);
  var num3 = parseInt(document.getElementById('num3').value);
  var sum = num1+num2+num3;
  var avg = sum/3;
  alert(avg);
}
</script>
```

תרגיל מס' 4

רמה: קלה

קילטו מן המשתמש שם וגיל. הציגו למשתמש כמה שנים נותרו לו עד גיל 120 לנצל.

פיתרון:

```
<input type='text' id='name' />
<input type='number' id='age' />
<button id='show()'>Show It! </button>

<script>
function show(){
  var name = document.getElementById('name').value;
  var age = parseInt(document.getElementById('age').value);
  var sum = 120 - age;
  alert(sum);
}
</script>
```

תרגיל מס' 5

רמה: קלה

קילטו מן המשתמש 10 ציונים. הציגו את הציון הגבוה ביותר, את הממוצע. בנוסף, כיתבו הערכה מילולית לכל ציון. ידוע כי ציון מתחת ל60 הוא נכשל, בין 60 ל70 הוא 'מספיק', בין 70 ל80 זה 'כמעט טוב', בין 80 ל90 זה 'טוב' ומ90 והלאה זה 'מצוין!'.

```
<input type='number' id='num1' />
<input type='number' id='num2' />
<input type='number' id='num3' />
<input type='number' id='num4' />
<input type='number' id='num5' />
<input type='number' id='num6' />
<input type='number' id='num7' />
<input type='number' id='num8' />
<input type='number' id='num9' />
<input type='number' id='num10' />

<button id='show()'>Show It! </button>

<script>
function show(){
var num1 = parseInt(document.getElementById('num1').value);
var num2 = parseInt(document.getElementById('num2').value);
var num3 = parseInt(document.getElementById('num3').value);
var num4 = parseInt(document.getElementById('num4').value);
var num5 = parseInt(document.getElementById('num5').value);
var num6 = parseInt(document.getElementById('num6').value);
var num7 = parseInt(document.getElementById('num7').value);
var num8 = parseInt(document.getElementById('num8').value);
var num9 = parseInt(document.getElementById('num9').value);
var num10 = parseInt(document.getElementById('num10').value);

var sum = num1+num2+num3+num4+num5+num6+num7+num8+num9+num10;
var age = sum/10;
alert(sum);
}
</script>
```

תרגיל מס' 6

רמה: בינונית

קילטו מן המשתמש מספר והציגו בפניו את כל המספרים מ1 ועד המספר שהוא כתב.

פיתרון:

```
<input type='number' id='num' />
<button id='show()'>Show It! </button>

<script>
function show(){
var num = parseInt(document.getElementById('num').value);
for (var i=1; i<=num; i++){
alert(i);

}
}
</script>
```

תרגיל מס' 7

רמה: קשה

קילטו מן המשתמש 3 מספרים. הציגו באופן נפרד לכל מספר את המספר עד 0 (כלומר, אם המספר הוא 5 אז הפלט יהיה 5-4-3-2-1). השתמשו רק בלולאת FOR אחת! וכן גם בפונקציה שתעזור לכם לייעל את הקוד.

הפיתרון:

```
<input type='number' id='num1' />
<input type='number' id='num2' />
<input type='number' id='num3' />

<button id='show()'>Show It! </button>

<script>
function show(){
var num1 = parseInt(document.getElementById('num1').value);
var num2 = parseInt(document.getElementById('num2').value);
var num3 = parseInt(document.getElementById('num3').value);
for (var i=num1; num1>=1; i--){
alert(i);
}
for (var j=num2; num2>=1; j--){
alert(j);
}

for (var k=num3; num3>=1; k--){
alert(k);
}
}
</script>
```

תרגיל מס' 8

רמה: אתגר

בנו תוכנה אשר מיועדת לגננת בגן.

לגננת יש עוגה והיא רוצה לפרוס אותה לפרוסות. קילטו את מס' פרוסות העוגה של הגננת וכן גם את מס' הילדים בגן.

בידקו כמה פרוסות נשארו לאחר החלוקה וכמה פרוסות כל ילד קיבל. (רמז: השתמשו במודולו).

הפיתרון:

```
<input type='number' id='cake' placeholder='Cakes Count' />
<input type='number' id='child' placeholder='Children Count' />

<button id='show()'>Show It! </button>

<script>
function show(){
var cake = parseInt(document.getElementById('cake').value);
var children = parseInt(document.getElementById('children').value);

var slices = parseInt(cake / children); var left = cake % children;
alert(slices + ' , ' + left);
}
</script>
```

תרגיל מס' 9

רמה: אתגר

קילטו מהמשתמש שם של מדינה. הציגו כמה אותיות יש בשם של המדינה. (רמז: היעזרו בlength).

הפיתרון:

```
<input type='text' id='country' />

<button id='show()'>Show It! </button>

<script>
function show(){
var country = document.getElementById('country').value;
alert(country.length);
}
</script>
```

תרגיל מס' 10

רמה: אתגר

קילטו מהמשתמש את שמו ואת שם חברו. הציגו מי החבר המנצח לו יש יותר אותיות בשם.

הפיתרון:

```
<input type='text' id='name1' />
<input type='text' id='name2' />

<button id='show()'>Show It! </button>

<script>
function show(){
var name1 = document.getElementById('name1').value;
var name2 = document.getElementById('name2').value;
if (name1.length > name2.length){
alert(name1);
}

else { alert(name2);
}
}
</script>
```

הגעתם עד לפה? כל הכבוד!!!

אירועים (Events)

בשיעור זה אלמד על אירועים בjs.

למעשה, נושא ה'אירועים' מתחלק לעולם הלחיצות על אלמנטים. למדנו על נושא הלחיצות גם בשיעור אודות 'קלט משתמש' וכן גם בשיעור 'הפונקציות', אולם יש עוד סוגים שונים של תזמוני קוד בלחיצות או במעברי עכבר. כל העולם הזה נקרא אירועים.

האירוע הנפוץ ביותר הוא כמובן לחיצה ברורה וחדה - onclick . בה כבר נפגשנו, היא הראשונה והשכיחה:

```
<button onclick='tal()'>Click Me!</button>
```

כאמור, בעת לחיצה על הכפתור, הפעל קוד מסוים . תזמון קוד בלחיצה.

אירוע נוסף הוא במעבר עכבר כמו Hover בעולם הcss. גם כאן, בjs,ניתן לתזמן קריאה לקוד כלשהו או פונקציה כלשהי במעבר עכבר. קרי, פתיחת תפריט צף במעבר עכבר על מילה כלשהי או על אייקון ההמבורגר במעלה הדף. איך נעשה זאת? - על ידי החלפה של onclick בonmouseover.

```
<button onmouseover='tal()'>Click Me!</button>
```

כאמור, כאשר המשתמש יזיז את העכבר מעל האלמנט ישר הפונקציה tal תצא לדרך.

אירוע נוסף הוא קריאה לקוד מסוים כאשר אתה יוצא מהאלמנט. למשל, האלמנט שלנו הוא div מרובע. רק כאשר נצא ממנו, נהיה מעוניינים שהוא יעלם... איך נעשה זאת? - על ידי החלפה של onmouseover בonmouseout.

```
<button onmouseout='tal()'>Click Me!</button>
```

נסו בעצמיכם! נסו לגרום לתפריט להופיע ואז להעלים את התפריט שיצרתם ! (רמז: דרך visible בcss).

לסיום, יש גם את onload. תרשמו בתוכו את הפונקציה הרלוונטית שתתקיים ברגע שהדף יעלה. למשל, אם ברצוננו לגרום להודעה קופצת איך שהאתר נוצר! נוכל לעשות זאת באמצעות הonload:

```
<body onload='alert('test')'></body>
```

ניתן גם לייצר אירועים שונים לא רק דרך התגית עצמה כפי שעשינו עד כה. ניתן גם דרך פונקציה גדולה וראשית. הדבר מעניק לנו סדר בעבודה וכתיבה של כל קודי הjs תחת תגיות הscript. שימו לב לדוגמא הבאה:

```
<div id='tal'> </div>
<script>
function tal() {
document.getElementById('demo').innerHTML = 'Hello World';
}
</script>
```

שינוי CSS דרך JS

בשיעור זה נבין כיצד ניתן לשחק עם הגדרות הcss באמצעות כתיבת סקריפטים בjs. למעשה, זהו נושא חשוב מאוד, משום שבאמצעות הcss נוכל לגרום להעלמת תגיות ולהצגתן (באמצעות התכונות display:none מול display:block), נוכל לגרום לטקסט מסוים להופיע על גבי תמונה בעת מעבר עכבר ועוד שלל תכונות שלא קיימות בcss הרגיל שנחשפנו אליו בחלק הקודם (ראה ערך: מדריך CSS)

הכתיבה הנכונה לשינוי תכונת css כלשהו בתגית קיימת היא תוספת של תכונת הcss שלנו מיד לאחר הרפרנס לid שלנו: `document.getElementById('id')`, למעשה הדבר ייראה כך בדוגמא הבאה למשל (שינוי צבע הכיתוב לירוק):

```
<script>document.getElementById('id').style.color = 'green';</script>
```

כאמור, איך שהאתר יוצג אוטומטית הטקסט יהיה בצבע ירוק.

בשיעור הקודם למדנו על אירועים ושם נוכל בהחלט לשלב את הקוד הנ'ל בתוך לחיצה למשל או מעבר עכבר...

```
<button onmouseover='tal()'>Click Me!</button>
```

נוכל לבצע מעבר עכבר על תגיות ולהעלים אותן...

```
<h1 id='onme' onmouseover='tal()'>עלה עליי</h1>
<script>function tal(){
document.getElementById('onme').style.display='none';
}
```

בזו הדרך נוכל לייצר תפריטים נפתחים למשל... פירוש הדבר הוא שהdiv יהיה בהתחלה במצב של display:none ועם מעבר העכבר הוא יהפוך לdisplay:block.

כמו כן, לגרום לטקסט להופיע על התמונה במעבר עכבר:

```
<div class='container'>
<div onmouseover='overit()' onmouseout='outit()' style='position: relative;
width:250px; text-align: center;'>
<img id='imgy' src='https://tal-web.co.il/images/android.jpg' height='150px'
width='250px'>
<div id='txty' style='display:none; position: absolute; top: 50%; left: 50%;
transform: translate(-50%, -50%);'>Centered</div>
</div>
</div>
<script>
function overit(){
document.getElementById('txty').style.display = 'block';
document.getElementById('imgy').style.filter = 'contrast(0.5)';
}
function outit(){
document.getElementById('txty').style.display = 'none';
document.getElementById('imgy').style.filter = 'none';
}
</script>
```

שמירת נתונים בצד לקוח – local Storage

הפעם נלמד איך לשמור מידע של המשתמש שלנו, כמו שם משתמש וסיסמא, ישירות אל הזיכרון של האתר שלנו.

בכך למעשה, האתר 'יזכור' מידע מסוים שהמשתמש הזין לתוכו ובפעם הבאה שהמשתמש יכנס לאתר, האתר כבר יכיר ויזכור אותו.

את הטכניקה הזו אנחנו מבצעים באמצעות אובייקט בשם localStroage והשמירה תתבצע על גבי הדפדפן הספציפי של המשתמש.

כך שבמידה והמשתמש שלכם יתחבר דרך דפדפן אחר או מחשב אחר, הוא לא יקבל את אותו המידע ששמר.

בואו נתרגל שמירה של שם משתמש.

צרו באתר שלכם h1 תגית ,input אחד עם `id = 'username'` וכפתור אחד.
בתוך ה h1 תנו את ההנחייה הבאה 'אנא הזינו את שמכם בתוך ה input ולסיום ליחצו על הכפתור.'

בעת לחיצה על הכפתור, בצעו את הפקודה הבאה:

```
<script>
function saveData(){
    var username = document.getElementById('username').value;
    localStorage.setItem('username', username);
}
</script>
```

המשמעות כאן היא לשמור מגירה חדשה בשם username ובתוכה לשים את הערך שיש בתוך הInput. במעין שיטת הkey והvalue.

איפה זה נשמר כך שנוכל לראות בעין שלנו?

1. ליחצו על רקע האתר באמצעות המקש הימני בעכבר

2. ליחצו על האפשרותinspect

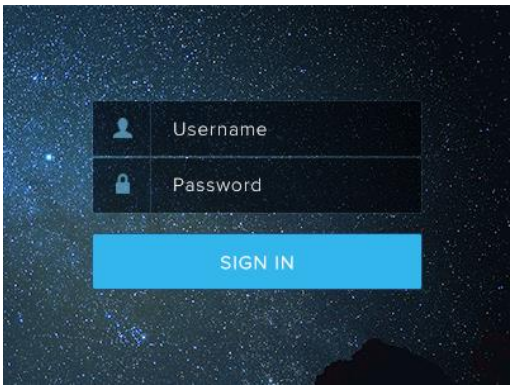
3. בחלונית שנפתחה לכם לחצו על הלשוניתApplication

4. בתוכה, יש מספר אפשרויות. אחת מהן היא Local Storage. ביחרו בה. שם יש מספר שורות ריקות, אבל הראשונה מביניהן תהיה הusername שאותו הוספתם בלחיצת כפתור.

כעת עולה השאלה איך ניתן גם למשוך את המידע.

ברגע שנצא ונכנס מהאתר או אפילו נעשה רענון, נרצה שתגית הh1 תשנה את הכיתוב שלה. תחילה היה כתוב בה את ההנחיות להוסיף את הטקסט החדש. אבל עתה, נרצה שיהיה כתוב בה 'ברוך שובך,' ואת השם של המשתמש! איך עושים זאת? באמצעות שורת קוד אחת שמוסיפים בתחתית הbody שלכם –

```
<script>
document.getElementById('h1Title').innerHTML =
'ברוך שובך,' + localStorage.getItem('username');
</script>
```



נספח 1 - jQuery

בדומה לbootstrap בשלב הcss, גם כאן קבוצת מתכנתים כתבו עבורינו שיפורים ופיצ'רים לjs. למעשה, אנו יכולים להתבסס על פונקציות שהם כתבו עבורינו ולתכנת יחד עם זה. במקומות עבודה רבים, לא מתכנתים על js טהור כפי שהראתי לכם עד כה, אלא משתמשים בjQuery או בספריות הנשענות על הjs.

ההבדלים בין js לjQuery מבחינת קלט משתמש: במקום הקריאה הארוכה ממקודם

```
document.getElementById('tal')
```

נוכל לרשום בהתבסס על הjQuery:

```
$('#tal')
```
בלבד.

כל שעלינו לעשות זה להוסיף את הסקריפט (ניתן גם להוריד אותו למחשב) שכתבו עבורינו עם שלל הפונקציות והקריאות השונות –

```
<script src='https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js'></script>
```

ובתוך תגית הסקריפט שלנו, נצטרך לעטוף את שלל קטעי הקוד השונים על ידי הקוד הבא:

```
<script>
$(document).ready(function(){
$('#tal').click(function(){

});
});
</script>
```

הקוד כתוב כך שבלחיצה על האלמנט עם הid של tal , יבצע כל הקוד הכתוב הפונקציה עצמה.
שימו לב, יש עולם שלם לjQuery עם כתיבה בשורות קוד מקוצרות ובפונקציות שונות... למשל, אפשרויות לגרור אלמנטים, אפשרויות לצבוע ולעצב בjQuery, אפשרויות לבצע התאמות רספונסיביות וקטעי קוד. למעשה, ניתן לומר כי צריך מדריך שלם רק על jQuery. גם בשני השיעורים מתחתיי אנסה להמשיך ולהרחיב עבורכם.

נספח 2 - jQuery UI

בהמשך לשיעור הקודם, כעת נגלוש לתוך עולם הjQuery. אם תיכנסו לאתר של הjQuery, בו יש את כל קטעי הקוד והפונקציות שכתבו עבורינו, נוכל לראות שם שיש עוד איזור למעלה בשם ui. כלומר - חווית משתמש. כל מיני קטעי קוד ופונקציות הנוגעות רק לויזואליות ולנראות עבור המשתמשים. בשביל החלק הזה , נצטרך להוסיף עוד קריאה לסקריפט חיצוני של הjQuery. כאשר גם אותו ניתן לשמור אם תרצו בכך.

```
<script src='https://code.jquery.com/ui/1.12.1/jquery-ui.js'></script>
```

מה נוכל לבצע באמצעות הjQuery ui? למשל, לתת למשתמש שלנו אפשרות גרירה של div מסוים. כל שתצטרכו לעשות הוא להוסיף לdiv שלכם את התכונה הבאה:

```
$('div').draggable();
```

בנוסף לכך, ניתן גם לשנות צבע רקע למשל בצורת css דרך הjQuery:

```
$('div').css('background-color','red');
```

או בצורה של אנימציה (animate זו פונקציה שמקבלת שני פרמטרים. האחד, בתוך סוגריים מסולסלים, הוא צבע רקע והשני הוא סך זמן שהאנימציה תימשך):

```
$('div').animate({'background-color':'red'},2000);
```

ל jqueryשימושים רבים והם כולם באים לשרת אותנו על מנת להעניק חווית משתמש טובה יותר ונוחה יותר עבור המשתמשים שלנו.

נספח 3 - jQuery mobile

בשיעור זה אלמד אתכם איך בעזרת jQuery ניתן להוסיף Panel נפתח בלחיצה על כפתור.
עיקבו אחר ההוראות להלן ולבסוף תוכלו ליצור גם אתם בעצמיכם Panel נפתח בצד הדף.

---

1. הוסיפו את קריאות הscript וcss של עולם הjquery mobile:

```
<script src='https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js'></script>
<script src='https://code.jquery.com/mobile/1.5.0-alpha.1/jquery.mobile-1.5.0-alpha.1.min.js'></script>
<link href='https://code.jquery.com/mobile/1.5.0-alpha.1/jquery.mobile-1.5.0-alpha.1.min.css' rel='stylesheet'/>
```

2. נוסיף div בbody שלנו עם התכונות הבאות:

```
<div data-role='panel' id='defaultpanel' data-theme='b'></div>
```

ובתוכו נכיל את כל התפריט הצדדי שלנו, אותו נרצה להציג בצד הדף. מעין תפריט נפתח

3. נוסיף לחיצה על אלמנט בתוך הדף שלנו:

```
<div class='ui-content'>
<a href='#defaultpanel'>Open panel</a>
</div>
```

כעת, אתם יכולים ליצור תפריט מרחף בלחיצה על כפתור באתר שלכם באמצעות הjQuery.
מומלץ לחלק את הדף הBODY לשני חלקים. חלק אחד זה הdiv של הpanel והחלק שני יהיה הdiv של הui-content וכל האתר עצמו.