

מה שנצטרך לדעת על <=== Javascript

זוהי שפת תכנות שמשמשת בעיקר 'לדבר' עם הדפדפן, כלומר עם דף ה-HTML שלנו.

בדומה ל- C#, גם ב-Javascript נכתוב תוכניות - אפילו התחביר מאד דומה.

זה יאפשר לבנות אתר יותר דינמי, למשל מעבר תמונות בעזרת לחיצת העכבר, בדיקה מיידית של תקינות קלט מהמשתמש ועוד.

היכן יכתב הקוד ?

התוכנית שכתובה ב- Javascript תיכתב בתוך דף ה-HTML.

מבנה מיוחד שיקיף את הקוד יראה כך:

```
<script type="text/javascript">
```

פקודות התוכנית

·
·
·

```
</script>
```

כל הקטע הזה יכול להופיע בכל מקום בדף ה-HTML - לרוב זה מופיע בין תגיות ה-`head`.

דוגמה

```
<html>  
  <head>  
    <script>  
      window.alert("Hello World")  
    </script>  
  </head>  
  
  <body>  
    <h1>JavaScript in Head</h1>  
  </body>  
</html>
```

השתמשנו באובייקט שנקרא window ושמייצג את כל מהשמופיע בדף האינטרנט שלפנינו, ובאחת הפעולות שאפשר ליישם עליו: alert. כדי להציג משו על המסך.

נסו זאת בעצמכם.

כל מה שנמצא בין התגיות של <script> ו- </script> אינו נחשב לחלק מדף ה-HTML ולכן אין הוא מוצג על ידי הדפדפן.

האובייקט חלון (window) הוא האובייקט הראשי בשפה. כל יתר האובייקטים שנכיר הם צאצאים שלו.

בנוסף לפעולה: alert, קיימת גם פעולה בשם: prompt שאפשר להפעיל על חלון, וכן מאפיין שנקרא: status.

על מנת לבקש קלט מהמשתמש נכתוב:

window.prompt("message", "default value")
message היא הבקשה לקלט ו- default value יהיה בשימוש אם לא הוקש דבר.

צורת ההפעלה:

```
<html>
  <head>
    <script>
      name = window.prompt("Please enter your name", "Unknown");
      window.alert("Welcome to the club: " + name);
    </script>
  </head>

  <body>
    <h1>JavaScript in Head</h1>
  </body>
</html>
```

מספר דברים שניתן להסיק מהקוד: משפט בשפה מסתיים בנקודה פסיק. משתנים לא חייבים להגדיר. כאן השתמשנו במשתנה מחרוזתי בשם: name שיקבל את הקלט, אם יוקש, או "Unknown" אחרת. ניתן ורצוי להצהיר על משתנים על ידי מילת המפתח var, למשל:

```
var name;
```

שימו לב שאין ציון של סוג המשתנה.

כמו כן פעולת שירשור בין מחרוזות נעשית בדומה ל- C# בעזרת הפלוס (+)

חשוב לזכור - כל ערך שמוקש על ידי המשתמש, מתורגם לערך מחרוזתי.

אובייקט - מסמך (document) הוא צאצא של window - זהו מסמך ה-HTML והנה מספר שימושים:

מאפיינים:

document.body.style.backgroundColor
או בקיצור: document.bgColor (צבע רקע המסמך).

document.URL
זו כתובתו של המסמך (url)

document.title
תוכן כותרת העמוד

פעולה חשובה:

```
document.write("<img src='pic.jpg' /> picture");
```

זוהי כתיבה לתוך מסמך ה-HTML - בצורה כזו אפשר 'לשתול' HTML דינמי לתוך הדף.

למשל הנה דוגמה ליצירת טבלה פשוטה

```
<html>
<head >
  <title>
</title>
</head>
<body>
  document.write("<table border='4'><tr> </tr><td>da da </td> <td>HHHHHHH <
    / td><tr><td>da da </td> <td>ga ga </td> </tr> </table>")
  </script>
</body>
</html>
```

הנה טבלה מורכבת יותר שתראה בערך כך:

Table of Numbers

Col Head 1	Col Head 2	Col Head 3
Number 0 is:	1.000	1.000
Number 1 is:	2.218	2.218
Number 2 is:	33.000	33.000
Number 3 is:	114.940	114.940
Number 4 is:	5.000	5.000
Number 5 is:	33.000	33.000
Number 6 is:	114.980	114.980
Number 7 is:	5.000	5.000

הקוד

```
<html>
  <!-- http://www.java2s.com/Tutorial/JavaScript/0220__Array/OutputarrayelementinaHTMLtableformat.htm
-->

  <head>
    <title>Table of Numbers</title>
  </head>

  <body>
    <h1>Table of Numbers</h1>

    <table border="0">
      <script language="javascript" type="text/javascript">
        <!--
          var myArray = new Array();
          myArray[0] = 1;
          myArray[1] = 2.218;
          myArray[2] = 33;
          myArray[3] = 114.94;
          myArray[4] = 5;
          myArray[5] = 33;
          myArray[6] = 114.980;
          myArray[7] = 5;

          var myTable= "<table><tr><td style='width: 100px; color: red;'>Col Head 1</td>";
          myTable+= "<td style='width: 100px; color: red; text-align: right;'>Col Head 2</td>";
          myTable+="<td style='width: 100px; color: red; text-align: right;'>Col Head 3</td></tr>";

          myTable+="<tr><td style='width: 100px;          '>-----</td>";
          myTable+="<td style='width: 100px; text-align: right;'>-----</td>";
          myTable+="<td style='width: 100px; text-align: right;'>-----</td></tr>";
```

```

for (var i=0; i<8; i++) {
    myTable+="<tr><td style='width: 100px;'>Number " + i + " is:</td>";
    myArray[i] = myArray[i].toFixed(3);
    myTable+="<td style='width: 100px; text-align: right;'>" + myArray[i] + "</td>";
    myTable+="<td style='width: 100px; text-align: right;'>" + myArray[i] + "</td></tr>";
}
myTable+="</table>";
document.write( myTable);
</script>
</table>
</body>

</html>

```

כאן גם ראינו שימוש במערך שלא היה צורך להגדיר את גודלו מראש.

פקודות התנאי if הן בעלות תחביר זהה לזה של C# וגם לולאות ה- for וה- while. פעולות החשבון, תנאי מורכב וביטויים חשבוניים ולוגיים גם הם כמו ב C#. משתמשים ב || עבור או, && עבור 'וגם'.

בכיתה עשינו תרגיל של תוכנית שקולטת 2 ציונים. אם שניהם 60 ומעלה, היא מודיעה: עברתם את 2 בחינות, אם רק אחד 0 ויותר, מודיעה שעברתם אחת, ואם שתין פחות מ-60 מודיעות על כישלון.

פתרון

```

<html>
<head >
<title>
</title>
</head>
<body>
<script type="text/javascript">

var grad1 = window.prompt("Enter the first grade: ");
var grad2 = window.prompt("Enter the second grade: ");
if (grad1 >= 60 && grad2 >= 60)
{
    alert("You have passed both exams");
}
else
{
    if (grad1 >= 60 || grad2 >= 60)
        alert("You have passed one exam only");
    else
        alert ("You failed both");
}

</script>
</body>
</html>

```

הוראת ה - switch הינה כמו ב-C#.

על מנת לעבוד עם מספרים מהקלט, נצטרך להסב את מחרוזת הקלט בעזרת: ParseInt או ParseFloat לדוגמה:

```
<html>
<head >
  <title>
</title>
</head>
<body>
  <script type="text/javascript">

    var num = window.prompt("Please enter a number between 1 and 3: ");
    num = parseInt(num)
    switch(num)
    {
      case 1: document.write("one");
              break;
      case 2: document.write("two");
              break;
      case 3: document.write("three");
              break;
      default: document.write("ERROR");
    }

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

על מנת לחשב ביטוי מחרוזתי ניתן להפעיל את הפעולה: eval למשל:
var x = eval("5+3"); ==> // x will have 8.

תרגילים

1.

כתוב תוכנית ב-Javascript שתיישם מחשבון של פעולות פשוטות (חיבור, חיסור, כפל וחילוק). התוכנית תבקש 2 מספרים ופעולה חשבונית (מסוג char) ותדפיס למסך את תוצאת התרגיל.

2.

כתוב תוכנית ב-Javascript שתדפיס את הטבלה הבאה:

Number	Square	Times 2	factorial
1	1	2	1
2	4	4	2
3	9	6	6
4	16	8	24

פונקציות

כמו ב- C# ישנן פונקציות שמקבלות ערכים ומחזירות ערך, או כאלה שמקבלות ולא מחזירות, או שלא מקבלות ומחזירו, או שלא מקבלות ולא מחזירות.

פונקציה מתחילה במילת המפתח: function ואחריה שם הפונקציה, סוגריים עבור פרמטרים וסוגריים מסולסלים עבור גוף הפונקציה.

הפונקציה לא מתבצעת אם לא קוראם לה (קטע קוד לא ביצועי). זימון הפונקציה נעשה כמו זימון פעולה סטאטית ב- C#.

דוגמה:

```
<html>
<head >
  <script type="text/javascript">
    function sample()
    {
      document.body.style.background="yellow";
    }
  </script>
</head>
<body>
  <script type="text/javascript">
    sample();
    document.write("<img src='nyc.jpg'/> pic1");
  </script>
</body>
</html>
```

זה יעבוד בצורה מלאה אם תהיה לכם תמונה שתיקרא: nyc.jpg בתיקיה שב שמור קובץ ה- HTML הזה. הפונקציה sample מזומנת מתוך הקוד שניכתב בתוך ה- body של ה- HTML (זוכרים קוד של Javascript יכול להופיע בכל מקום בדף בתנאי שהוא מוקף בתגיות <script> ו- </script>

```
<html>
  <head>
    <script type="text/javascript">
      function sayHello()
      {
        document.write ("Hello there!");
      }
    </script>
  </head>
  <body>
    <p>Click the following button to call the function</p>

    <form>
      <input type="button" onclick="sayHello()" value="Say Hello">
    </form>

    <p>Use different text in write method and then try...</p>
  </body>
</html>
```

כאן גם שתמשנו ב 'ארוע' שנקרא: onclick. יצרנו כפתור והאירוע הפעיל את הפונקציה על ידי לחיצה על הכפתור. נראה עוד כאלה בהמשך.

משתנים מקומיים וגלובאליים

משתנה מקומי מוגדר על ידי מילת המפתח var. אם הוא בפונקציה, אינו מוכר מחוצה לה.

משתנה גלובאלי מוגדר מחוץ לפונקציות והוא מוכר בכל הפונקציות שבאותו דף HTML.

דוגמה:

a הוא גלובאלי ו- b מקומי בפונקציה myvar.


```
<html>
  <head>
    <script type="text/javascript">
      var a = 5;
      function myvar()
      {
        var b = 7;
        alert (a*b);
      }
    </script>
  </head>
  <body>
    <script type="text/javascript">
      myvar();
    </script>
  </body>
</html>
```

התוצאה שיודפס הערך 35 בחלון.

אם נגדיר אותו שם של משתנה גם כגלובאלי וגם כמקומי, הפונקציה בה הוגדר כמקומי לא תשפיע על זה הגלובאלי.

תרגילים:

מאחד עד חמש בתחתית הדף הבא. (תרגילים לנושא 6 - פונקציות)

המשך הדיון על
onclick ואירועים דומים כמובטח:

תרגילים על אירועים ופונקציות:

תכנות בליטה מכוללת אירועים

JavaScript תומכת ב"תכנות מכוון אירועים". בדפדפן קיימים מספר אירועים שאותם הוא יכול לזהות ועבורם נוכל לרשום תגובה. כלומר נוכל לתכנת כך שאירוע ישמש כמפעיל **Trigger** לפעולה. כשהדפדפן יזהה אירוע מסוים תתבצע כתגובה פעולה הנמצאת בתסריט. אירוע יכול להיות פעולה של מערכת ההפעלה כגון שרון או פעולה שביצע המשתמש כגון: סגירת מסמך, מעבר עכבר, לחיצת עכבר, לחיצת כפתור, בחירה באזור מסוים וכו'. תגובה לאירוע יכולה להתבצע בזימון פונקציה או ביצוע קוד Script.

רשימת אירועים – event חלקית לפי סדר א"ב:

אירוע מפעיל Trigger	פועל כאשר	פועל על
onblur	שדה בטופס מאבד מיקוד (focus). לדוגמה, מילוי נתון ומעבר למילוי נתון נוסף.	מרכיבי טופס
onchange	המשתמש משנה שדה בטופס. לדוגמה, ניתן לידע את המשתמש בכך שתוכן השדה שונה.	מרכיבי טופס
onclick	המשתמש לחץ בעזרת העכבר על רכיב בדף. הערה: לא פועל על frame.	תמונות, טקסט, שדות, כפתורים וכו'
onerror	יש בעיה בטעינת התמונה או בהרצת הקוד.	
onfocus	אובייקט נבחר. לדוגמה כניסה לשדה טקסט.	מרכיבי טופס
onkeypress	המשתמש לחץ על מקש כלשהו (לחיצה ושחרור).	
onload	מסתיימת טעינה של חלון או תמונה.	body
onunload	המשתמש מבקש לעזוב את הדף.	body
onmouseover	מצביע העכבר מצביע על עוגן או על אובייקט.	
onmouseout	מצביע העכבר יוצא מקישור או אובייקט.	
onreset	המשתמש לוחץ על לחצן Reset.	
onselect	כאשר נבחר שדה בטופס.	מרכיבי טופס
onsubmit	נלחץ לחצן Submit לשליחה של טופס.	לחצן Submit
onresize	משנים את גודל החלון.	body

הערה: רשימת האירועים המפעילים (Trigger) שייכת ל-HTML. אין משמעות לגודל האותיות: **OnClick** שקול ל **onclick**.

func7.htm

תרגיל 7: שינוי צבע הרקע בלחיצת כפתור

- כתוב פונקציה המקבלת צבע ומשנה את צבע הרקע של המסמך בהתאם.
- צור 3 כפתורים ועליהם רשומים צבעים שונים. לחיצה על כל כפתור תשלח לפונקציה שרשמת בסעיף א' שתשנה את הצבע בהתאם למה שרשמת.

func8.htm

תרגיל 8: שינוי צבע הרקע במעבר עכבר

- צור מסמך html והוסף בו תמונה במיקום כלשהו.
 - שנה את צבע הרקע של המסמך לצהוב במעבר עכבר על התמונה (onmouseover).
 - שנה את הצבע לירוק - ביציאה מתחום התמונה (onmouseout).
- שים ♥ : היעזר בפונקציה שיצרת בתרגיל 7 א' ובמבנה הבא:
- ```

```

func9.htm

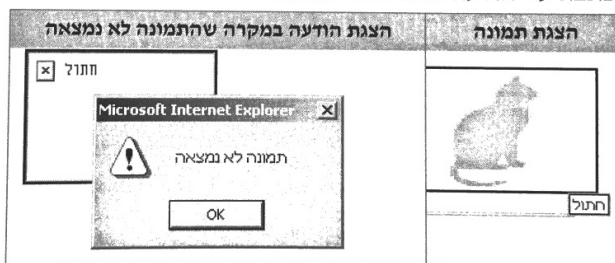
### תרגיל 9: שינוי צבע הרקע במעבר עכבר

- צור מסמך html המכיל את האפשרות לשינוי גודל הדף (בעזרת האירוע OnResize) וכן הצגה של הודעה "שונה גודל הדף" (הנחיה: רשום את האירוע בתוך התג body).

func10.htm

### תרגיל 10: שינוי צבע הרקע במעבר עכבר

- בנה פונקציה myAlert(mes) המקבלת הודעה במשתנה mes ומדפיסה את ההודעה ב-alert.
- בנה מסמך html המציג תמונה (קבע גובה, רוחב מסגרת ותגית alt המכילה שם תמונה).
- בדוק אם קיימת בעיה בטעינת התמונה בעזרת onerror. אם כן, זמן את הפונקציה myAlert שכתבת עם הודעה מתאימה. דוגמה:



func11.htm

### תרגיל 11: "נחש מספר"

- צור משחקון בו צריך המשתתף לנחש מספר (ניתן לקבוע במשתנה את המספר שצריך לנחש ע"י הוראת השמה).
- כללי המשחק:

- המשתמש צריך לנסות לנחש מספר;
- מותר לנחש את המספר לכל היותר 6 פעמים;
- לאחר כל ניחוש שגוי מתקבלת הודעה: "המספר שבחרת גדול מדי" או "המספר שבחרת קטן מדי";
- כאשר המשתמש מצליח לנחש את המספר התכנית תעצור ותתקבל הודעה "כל הכבוד!!!".

## ז'מון פונקציה בעזרת תגית קישור

ניתן לזמן פונקציה בעזרת תגית קישור והמאפיין **href**. בזימון בעזרת תגית קישור חייבים לרשום את המילה **JavaScript** נקודותיים ואחייב את שם הפונקציה אותה מזמנים- ולתחום את הכול בגרשיים.



שם הקובץ באתר ExampleFun6\_6

### דוגמה פתורה 6.6: זימון בעזרת קישור

כתוב פונקציה המשנה את צבע הרקע לאדום.

כתוב פונקציה המשנה את צבע הרקע לכחול. זמן את כל אחת מהפונקציות בעזרת תגית קישור.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>&lt;html&gt; &lt;head&gt;   &lt;script type="text/javascript"&gt;     function bgred()     {       document.body.style.backgroundColor = "red";     }     function bgblue()     {       document.body.style.backgroundColor = "blue";     }   &lt;/script&gt; &lt;/head&gt; &lt;body&gt;   &lt;a href="javascript: bgred()"&gt;שנה רקע לאדום&lt;/a&gt;   &lt;a href="javascript: bgblue()"&gt;שנה רקע לכחול&lt;/a&gt; &lt;/body&gt; &lt;/html&gt;</pre> | <p>פונקציות לשינוי צבע רקע של המסמך</p> <p>זימון בעזרת תגית קישור<br/><b>התוצאה בדפדפן:</b></p> <p>שנה צבע רקע לאדום<br/>שנה צבע רקע לכחול</p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|

## תרגילים אלוטא 6: פונקציות

כתוב פונקציה

שמור את התרגילים הבאים בתיקיה jsCh6Function. הוסף עבור כל תרגיל קישור מהדף הראשי

1. כתוב פונקציה המקבלת כפרמטר 2 מספרים ומציגה הודעה מי מהם הוא הגדול. *func1.htm*
2. כתוב פונקציה המקבלת כפרמטר 3 מספרים ומציגה את ערכו של הגדול. *func2.htm*
3. כתוב פונקציה המקבלת כפרמטר מחרוזת המייצגת צבע חוקי ומשנה את רקע המסמך בהתאם לצבע שקיבלה. *func3.htm*
4. א. כתוב פונקציה המקבלת כפרמטר מספר ומחזירה את הערך 0 אם המספר שקיבלה זוגי אחרת תחזיר 1.  
ב. כתוב תכנית הקולטת מספר בעזרת `window.prompt` ובודקת בעזרת הפונקציה שכתבת בסעיף א' האם המספר זוגי או לא.
5. א. כתוב פונקציה המקבלת שני מספרים ומחזירה את מכפלתם \* 2. (השתמש בסעיף א').  
ב. כתוב תכנית הקולטת שני מספרים ומציגה הודעה מהי מכפלתם \* 2.



הצגת תמונות מתחלפות במעבר העכבר ויציאתו המיקום הנתון

```
<html>
<head>
 <script>
 function catpic()
 {
 pic.src="images/cat.gif"
 }
 function dogpic()
 {
 pic.src="images/dog.gif"
 }
 </script>
</head>
<body>
 <h1> Go over pic </h1>

</body>
</html>
```

**נושא 8**

**טיפול בטופס צד לקוח**



**מהו טופס?**

נציג כאן שתי הגדרות שישרתו את המטרות של כתיבת הפרק:

**הגדרה 1:** הטופס הוא כלי לאיסוף נתונים הבנוי משדות אותם ממלא הגולש. נציג קבלת הנתונים בשרת דואר, בדפי שרת ושמירתם במסד נתונים (הרחבה בנושא 9: טפסים – עמוד 57).

**הגדרה 2:** הטופס הוא אובייקט השייך לאובייקט מסמך שנגזר מאובייקט חלון (ראה תרשים בנושא 2: אובייקט חלון בעמוד 83).

שתי ההגדרות יתנו מענה הולם להתייחסות לטופס, הן מבחינת הגולש (שממלא את הנתונים בטופס או בוחר אותם מתוך רשימה) והן מבחינת המתכנת שבונה את הטופס וצריך לבדוק את תקינותו.

**טעם...** בעבר כאשר הלקוח היה ממלא טפסים, בדיקת אימות הנתונים של הטופס הייתה מתבצעת רק בשרת. הלקוח היה ממלא את הטופס ושולח, בשרת היו מתבצעות בדיקות כגון האם מולאו כל השדות, האם כתובת הדואר חוקית וכו'. תוצאות הבדיקה היו חוזרות ללקוח עם דרישה לתיקון. הליך זה היה מאט את קצב העבודה אצל הלקוח וגורם עומס מיותר על השרת.

**כיום...** לפני שליחת הנתונים נהוג לבדוק במחשב הלקוח את הערכים שהוקלדו בטופס ולבדוק שהטופס מולא כנדרש ככל שניתן. **בדיקת הטופס בצד לקוח אינה מבטלת את הצורך בבדיקות גם בצד שרת אך מצמצמת את מספרן של הבדיקות החוזרות. את הבדיקות בצד לקוח נבצע בעזרת הוראות, אובייקטים ופעולות ב-JavaScript.**

**בדיקת שדות טופס ואימות נתונים לשם מה?**

טופס הוא כלי חשוב ושכיח לאיסוף נתונים מהגולשים ומשתמשים באתרי אינטרנט רבים.

**דוגמאות לבדיקות בטופס**

- בשדה ת.ז. – האם נרשם ערך המורכב מ-9 תווים.
- בשדה ד.אלקטרוני – האם כתובת הדואר חוקית (למשל האם הכתובת מכילה @).
- בשדה מיקוד – האם מספר הספרות 5.

בחלק ב' של הספר: html למדנו כיצד לבנות טפסים ולשלבם בדפי האתר.

בחלק ג' (הנוכחי) של הספר: JavaScript נלמד כיצד לבדוק את הטפסים שמילאו הגולשים בעזרת

פנייה לערכים שבשדות הטופס. הפנייה מתבצעת בעזרת תחביר הנקודה. **שם שדה.שם טופס. document**

(הערה: התחביר המלא הוא -...שם שדה.שם טופס. windows. document)

בפרק הבא נראה כיצד ניתן לבצע על הערכים בדיקת מורכבות יותר בעזרת פונקציות של מחרוזות.



## בדיקת לדה טקסט

שדה טקסט הוא שדה בו הגולש מקליד מחרוזת תווים. את הערך בשדה טקסט ניתן לשלוח ולהתייחס אליו כאל מחרוזת טקסט ולבצע עליו את כל הפעולות שביצענו על טקסט כגון: השוואה, הדפסה, השמה במשתנה וכו'.

### מבנה הפנייה לערך שדה טקסט בטופס

document.value שם השדה בטופס. שם הטופס.

פנייה לשדה טקסט

document.value= ערך שם השדה בטופס. שם הטופס.

השמת ערך בשדה טקסט



דוגמה פתורה

שם הקובץ באתר ExampleForm8\_1

### דוגמה פתורה 8.1: שדה טקסט

בנה טופס בו יקליד הגולש את צבע הרקע של המסמך או יקליד קוד הקסדצימלי של הצבע. לאחר אירוע לחיצת כפתור ישתנה צבע הרקע בהתאם לערך שהוקלד.

פתרון

התוצאה בדפדפן	בניית טופס המכיל שדה טקסט וכפתור
<p>הפונקציה myBg משנה את הרקע במסמך בהתאם לערך שיוקלד בשדה. בפונקציה בדיקה הוראה אם הוקלד ערך. אם לא הוקלד ערך יופיע חלון ובו כתוב "הכנס צבע". אם הוקלד ערך תשנה הפונקציה את צבע הרקע בהתאם לערך שהתקבל מהטופס.</p>	<pre> &lt;html &gt; &lt;head&gt; &lt;title&gt; טפסים &lt;/title&gt; &lt;script type="text/javascript"&gt; function myBg () {     לקבלת הערך שהוקלד בשדה הטקסט     if (document.change.color.value == "")         alert("הכנס צבע");     else     {         var c = document.change.color.value;         document.body.style.backgroundColor=c;         document.change.color.value="בוצע";     } } &lt;/script&gt; &lt;/head &gt; &lt;body&gt;     הקש שם של צבע או ערך הקסדצימלי     &lt;form name="change"&gt;         &lt;input type="text" name="color" size="10" &gt;&lt;br /&gt;         &lt;input type="button" value="לחץ לשנות צבע" onclick="myBg()" &gt;     &lt;/form&gt;     הפעלת הפונקציה לשינוי צבע     לאחר האירוע OnClick. &lt;/body&gt; &lt;/html &gt;                 </pre>
<p><b>הטופס כפי שמוצג לגולש:</b></p> <p>הקש שם של צבע או ערך הקסדצימלי</p> <p><input type="text"/></p> <p>לחץ לשנות צבע</p> <p><b>הטופס לאחר הקלדת צבע רקע:</b></p> <p>הקש שם של צבע או ערך הקסדצימלי</p> <p>gold <input type="text"/></p> <p>לחץ לשנות צבע</p> <p><b>לאחר הלחיצה על הכפתור:</b></p> <p>הקש שם של צבע או ערך הקסדצימלי</p> <p>בוצע <input type="text"/></p> <p>לחץ לשנות צבע</p>	



## בריקת לדה רדיו

שדה רדיו הוא שדה בו הגולש מסמן אחת מבין מספר אפשרויות המוצגות לפניו ככפתורי רדיו ☉ בטופס המכיל כפתורי רדיו יש לכל האיברים (כפתורי הרדיו באותו הבלוק) שם זהה. זהו למעשה מערך\* בעל שם אחד המכיל ערכים שונים. ניתן לגשת לכל אחד מערכי המערך באמצעות מיקומם. \* מערך הוא אוסף ערכים מאותו טיפוס שהגישה אליהם ע"י ציון שם המערך ומיקום האיבר

### מבנה הפניה לערך שדה רדיו בטופס

`document.שם הטופס.value` [מיקום הכפתור במערך] שם השדה בטופס. שם הטופס.



דוגמה פתורה

דוגמה פתורה 8.2: כפתורי רדיו שם הקובץ באתר ExampleForm8\_2

בנה טופס בו הגולש מסמן אחת מתוך 3 אפשרויות המציננות את השעה שהגיע לבית המלון. לאחר אירוע של לחיצת כפתור תוצג בחלון הודעה עם הבחירה.

### פתרון

קוד התכנית - בניית טופס המכיל שדה רדיו וכפתור (הסבר מפורט ראה עמוד הבא)	התוצאה בדפדפן
<pre> &lt; html dir="rtl"&gt; &lt; head &gt; &lt; title &gt; כפתורי רדיו - טפסים &lt;/ title &gt; &lt; script type="text/javascript"&gt; function bdika() { for(i=0;i&lt;document.hotel.reaching.length;i++) { if(document.hotel.reaching[i].checked==true) { alert(document.hotel.reaching[i].value); break; } } } &lt;/ script &gt; &lt;/ head &gt; &lt; body &gt; &lt; form name="hotel"&gt; בחר שעת הגעה &lt; br /&gt; &lt; input type="radio" name="reaching" value="morning" /&gt; שעות הבוקר &lt; input type="radio" name="reaching" value="noon" /&gt; שעות הצהריים &lt; input type="radio" name="reaching" value="evening" /&gt; שעות הערב &lt; input type="button" value="בדיקה" onclick="bdika()" /&gt; &lt;/ form &gt; &lt;/ body &gt; &lt;/ html &gt;                 </pre> <p>הפונקציה מזהה את כפתור הרדיו שהגולש בחר ומציגה את בחירתו בחלון.</p> <p>שם הטופס</p> <p>להפרדת שורות &lt;br /&gt; הוסף</p>	<p><b>הטופס כפי שמוצג לגולש:</b></p> <p>בחר שעת הגעה</p> <p><input type="radio"/> שעות הבוקר</p> <p><input type="radio"/> שעות הצהריים</p> <p><input type="radio"/> שעות הערב</p> <p><input type="button" value="בדיקה"/></p> <p><b>הטופס לאחר בחירת כפתור:</b></p> <p>בחר שעת הגעה</p> <p><input checked="" type="radio"/> שעות הבוקר</p> <p><input type="radio"/> שעות הצהריים</p> <p><input type="radio"/> שעות הערב</p> <p><input type="button" value="בדיקה"/></p> <p><b>לאחר הלחיצה על הכפתור "בדיקה":</b></p>



הסבר:

1.

איך ניתן לדעת כמה כפתורי רדיו יש?  
 לאוסף כפתורי הרדיו יש מאפיין שנקרא `length` (אורך) המאפשר לדעת כמה כפתורים הוגדרו.  
 אם נרשום `length` (אורך) `document.hotel.reaching` נקבל את המספר 3, יש 3 איברים.  
 מאפיין אורך



2. כיצד נדע מה הגולש בחר ?

כל כפתור רדיו יש מאפיין שנקרא `checked`. ערכו של מאפיין `checked` הוא `true` אם הכפתור נבחר וערכו `false` אם הכפתור לא נבחר. כדי לדעת אם כפתור נבחר יש לעבור על כל הכפתורים ולבדוק עבור כל כפתור את ערך המאפיין `checked`. הכפתור שעבורו המאפיין `checked` הוא `true` הוא הנבחר.

3. כיצד נגיע לערך של כל איבר ?

מערך כפתורי הרדיו מכיל 3 איברים. האיברים ממוקמים מהמקום ה-0 עד מקום 2.

מיקום	הערך
0	morning
1	noon
2	evening

בחר שעת הגעה

שעות הבוקר ←  
 שעות הצהריים ←  
 שעות הערב ←

נוכל להגיע ולהציג את ערכו של כל איבר על-ידי ציון מיקומו ושימוש במאפיין `value`. למשל:  
`document.hotel.reaching[0].value`  
`document.hotel.reaching[1].value`

4. לולאת ה- `for` לסריקת כפתורי הרדיו ומציאת הערך שסומן :

הלולאה צריכה להתבצע החל מ-0 (מיקום האיבר הראשון) ועד למיקום האיבר האחרון.  
 בגוף הלולאה נבדק ערכו של המאפיין `checked`. אם ערכו `true` אז מצאנו את הכפתור הנבחר ונציג את ערך הכפתור בחלון.  
 ניתן לסיים את הלולאה לאחר שנמצא הכפתור (הערך) באמצעות הוראת `break`. שימוש בהוראה `break` אינו חובה אך מאפשר לתכנית להיות יעילה יותר.

### בדיקת לוגיקת רשימה נגללת

רשימה נגללת היא שדה המאפשר לגולש לבחור אחד או כמה מבין הפריטים הרשומים בה.

מבנה הפנייה לערך של שדה ברשימה נגללת בטופס

`document.selected` שם השדה בטופס. שם הטופס. `document`

מיקום האיבר הנבחר מהרשימה

`document.options[מיקום].value` שם השדה בטופס. שם הטופס. `document`

ערך האיבר הנבחר מהרשימה



דוגמה פתורה

שם הקובץ באתר ExampleForm8\_3

דוגמה פתורה 8.3: רשימה נגללת

בנה טופס המכיל רשימה נגללת עם שלושה שמות. לאחר אירוע של לחיצת כפתור יוצג בחלון השם.

בניית טופס המכיל רשימה נגללת וכפתור	התוצאה בדפדפן
<pre> &lt;html dir="rtl"&gt; &lt;head&gt; &lt;script type="text/javascript"&gt; function bdika() {   var mikum ;   mikum = document.tofes.MYlist.selectedIndex ;   // מיקום=(אינדקס) הערך שנבחר מתיבת הבחירה ;   alert(document.tofes.MYlist.options[mikum].value) ;   // הערך= (תוכן) הנבחר בתיבת בחירה } &lt;/script&gt; &lt;/head&gt; &lt;body &gt; &lt;form name="tofes"&gt;   &lt;select name="MYlist" &gt;     &lt;option value="Ron"&gt; רון &lt;/option&gt;     &lt;option value="Ben"&gt; בן &lt;/option&gt;     &lt;option value="Bar"&gt; בר &lt;/option&gt;   &lt;/select&gt;   &lt;input type="button" value="בדיקה" onclick="bdika()"&gt; &lt;/form&gt; &lt;/body&gt; &lt;/html&gt; </pre>	<p>הטופס כפי שמוצג לגולש:</p> <p>הטופס לאחר פתיחת הרשימה:</p> <p>לאחר הלחיצה על הכפתור "בדיקה":</p>

הסבר:

1. כיצד נדע מיקום האיבר הנבחר מהרשימה?

לרשימה נגללת יש מאפיין שנקרא selectedIndex. המאפיין מכיל את מיקומו של הערך הנבחר למשל: אם נבחר האיבר הראשון ערכו של המאפיין יהיה 0. נוכל להשיג את הערך המתקבל במשתנה. למשל: `mikum=document.tofes.MYlist.selectedIndex`.



2. כיצד נגיע לערך של כל איבר? בדוגמה יש 3 איברים המסודרים כמערך. האיברים ממוקמים החל מהמקום ה-0 עד מקום 2. נוכל להגיע למיקום הערך הנבחר ע"י ציון מקומו והמאפיין.

ערך מוצג	מיקום	הערך
רון	0	Ron
בן	1	Ben
בר	2	Bar

נוכל לקבל את ערכו של כל איבר על-ידי ציון מיקומו ושימוש במאפיין value באופן הבא:

מיקום במערך    שם השדה    שם הטופס  
`document.tofes.MYlist.options[mikum].value`  
 נוכל להציג את הערך ולבצע עליו פעולות, למשל:  
`alert(document.tofes.MYlist.options[mikum].value)`







שם הקובץ באתר ExampleForm8\_4

### דוגמה פתורה 8.4: רשימה נגללת

בנה טופס המכיל אפשרויות לבחור בין 3 סוגי סרטים. הבחירה הראשונה בטופס תציג ערך "בחר סוג". לאחר בחירת הסרט, יוצג למשתמש שם הסרט שבחר. אם לא נבחר סרט תירשם הודעה "בחר סוג".

דרך א: בחירה באחת האפשרויות המוצגות ברשימה תזמן את פונקציה הבדיקה

```

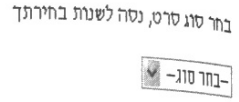
<html dir="rtl">
<head>
<script type="text/javascript">
<!--
function bdika() ←
{
 var mikum ;
 mikum = document.tofes. movie.selectedIndex ;
 // מיקום= (אינדקס) הערך שנבחר מתיבת הבחירה
 if (mikum ==0)
 // אם ערך הבחירה 0 אז נבחרה האפשרות הראשונה שבה לא הצבנו ערך
 alert ("בחר סוג סרט");
 else
 alert (document.tofes. movie.options[mikum].value) ;
 // הערך= (תוכן) הנבחר בתיבת בחירה
}
//-->
</script>
<title>
</title>
</head>
<body>
<div align="rtl">
<form name="tofes">
 <select name="movie" dir="rtl" onchange="bdika()">
 <option value="no_select" > --בחר סוג-- </option>
 <option value="comedy" > קומדיה </option>
 <option value="drama"> דרמה </option>
 <option value="documentary"> דוקומנטרי </option>
 </select>
</form>
</div>
</body>
</html>

```

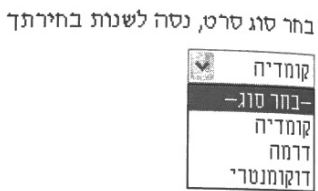
הפונקציה bdika בודקת את מיקום הערך הנבחר. אם מיקומו 0 אזי לא נבחר סוג, אחרת יוצג סוג הסרט בחלון.

בחירת אפשרות מהרשימה (onchange) תזמן פונקציה bdika()

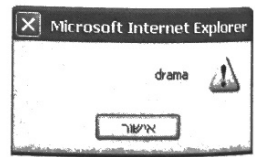
הטופס כפי שמוצג לגולש:



הטופס לאחר פתיחת הרשימה:



לאחר הלחיצה על הכפתור "בדיקה":



דרך ב': יש להגדיר כפתור "בדיקה". לאחר הבחירה באחת האפשרויות המוצגות ברשימה יש לחוץ על הכפתור לזימון פונקציה הבדיקה.





## בדיקת לוגי תיבת סימון - checkbox

בתיבת סימון ניתן לסמן מספר ערכים. לכל תיבה שם שונה וערך שונה. לבדיקת הערכים שבחר המשתמש נבדוק אם השדה נבחר ורק אח"כ נציג את הערך. בדיקה האם שדה מסוים נבחר, ערך המאפיין checked הוא true

מבנה פנייה לערך בתיבת סימון checkbox

document.שם\_השדה בטופס.שם\_הטופס.checked == true

ערך איבר כלשהו מהרשימה



דוגמה  
פתורה

דוגמה פתורה 8.5: תיבת סימון שם הקובץ באתר ExampleForm8\_5

אורח בבית מלון צריך לסמן את הארוחות שברצונו לקבל. הטופס מכיל רשימה ובה 4 ארוחות, תיבות סימון וכפתור

התוצאה בדפדפן	בניית טופס המכיל תיבת סימון וכפתור
ציין אילו ארוחות ברצונך לקבל <input type="checkbox"/> בוקר <input type="checkbox"/> צהריים <input type="checkbox"/> ערב <input type="checkbox"/> לילה <input type="button" value="בדוק"/>	<pre> &lt;form name="hotel"&gt;   &lt;br /&gt; ציין אילו ארוחות ברצונך לקבל   &lt;input name="cbox1" type="checkbox" value="meal1" /&gt; בוקר &lt;br /&gt;   &lt;input name="cbox2" type="checkbox" value="meal2" /&gt; צהריים &lt;br /&gt;   &lt;input name="cbox3" type="checkbox" value="meal3" /&gt; ערב &lt;br /&gt;   &lt;input name="cbox4" type="checkbox" value="meal4" /&gt; לילה &lt;br /&gt;   &lt;input type="button" value="בדוק" onclick="bdika()" /&gt; &lt;/form&gt; </pre>

כתוב פונקציה המציגה את הארוחות שסימן הלקוח (זמן את הפונקציה בלחיצת כפתור). פתרון א':

לכל אחד מהכפתורים שם שונה: cbox1: מייצג ארוחת בוקר וערכו meal1 וכ"ו.. אם נרצה להציג את כל הערכים שנבחרו, נבדוק עבור כל שדה האם הוא נבחר, אם כן נציג את ערכו

function bdika( )

```

{
 if (document.hotel.cbox1.checked==true)
 alert(document.hotel.cbox1.value);
 if (document.hotel.cbox2.checked==true)
 alert(document.hotel.cbox2.value);
 if (document.hotel.cbox3.checked==true)
 alert(document.hotel.cbox3.value);
 if (document.hotel.cbox4.checked==true)
 alert(document.hotel.cbox4.value);
}

```

מבנה הבדיקה עבור כל שדה: אם (הערך נבחר) הצג את ערכו

פתרון ב': ניתן בטופס לתת לכל האיברים של תיבות הסימון שם זהה ואז נוצר אוסף תיבות סימון שפנייה אליהם כמו אל אברי מערך\*, ואז הסריקה מתבצעת בלולאה כמו בכפתורי רדיו ע"י ציון מיקומם!!

\* מערך הוא אוסף ערכים מאותו טיפוס שהגישה אליהם ע"י ציון שם המערך ומיקום האיבר

document.hotel.cbox[0].value













