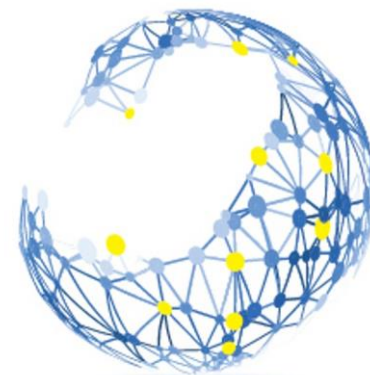


השתלמות מורים

תכנות בסביבת האינטרנט

גלעד מרקמן

קריית החינוך פארק המדע,
נס ציונה



קריית החינוך
השש שנתית
פארק המדע
עתיד של חדשנות



תכנות צד לקוח

JavaScript

- JavaScript היא שפת תכנות המיועדת לדפי אינטרנט, ונועדה לאפשר יצירת דפים דינמיים המגיבים לקלט מהמשתמש.
- שפת תסריט (Script) היא שפה המורצת באופן מיידי מהקוד, ואין צורך לבצע הידור (קומפילציה). שפת JavaScript מורצת על ידי הדפדפן ללא ביצוע של הידור קודם לכן.
- שפת JavaScript ושפת Java אילו הן שתי שפות שונות לחלוטין ואין להתבלבל ביניהם.

- נתן לכתוב את קוד JS בשני אופנים:
 - בתוך תגית `<script>` - Internal
 - בקובץ נפרד וקישור הקובץ לדף HTML – External - הדרך המומלצת.
- תגית `<script>` יכולה להיכתב הן בראש העמוד (בתוך תגית `<head>`) והן בגוף העמוד (בתוך תגית `<body>`).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>JS1</title>
    <script type="text/javascript">
      alert("Hello World");
    </script>
  </head>
  <body>
    <h1>JavaScript</h1>
  </body>
</html>
```

- כמקובל נתחיל בתוכנית Hello World
- הקוד נכתב בתגית `<script>`
 - הוספנו את המאפיין `type`. אין חובה להוסיף אותו.
- הפקודה `alert` יוצרת חלון התראה עם ההודעה הכתובה בסוגריים.
- כל פקודה ב JS מסתיימת בנקודה פסיק (;).
- הקוד מורץ מייד כשהדף עולה !!!

- כתיבת קוד בקובץ נפרד (External) נעשית לפי השלבים הבאים:
 - יצירת ספרייה מיוחדת לקבצי קוד
 - יצירת קובץ עם סיומת js. בתוך הספרייה.
 - כתיבת הקוד בקובץ החדש ללא צורך בתגית <script>.
 - קישור קובץ ה JS לקובץ HTML באמצעות הפקודה:

```
<script src="../../JS Code/JavaScript.js"></script>
```

- ניתן לקשר מספר קבצי קוד JS לדף אינטרנט.
- מקובל לכתוב את הקישור או הקוד (תגית ה <script>) בראש העמוד ולא בגוף.



פקודות בסיסיות

- בגאווה סקריפט ניתן להצהיר על משתנה בדרכים הבאות:

```
x = 5;           // global variable
var y = 5;       // function scope
let z = 5;       // block scope
const c = 3.14;  // block scope

var y1;
let z1;
```

- הסוג הראשון אינו מומלץ. יוצר משתנה גלובאלי.
- משתנה מסוג var הוא הנפוץ ביותר, שכן הסוגים האחרים קיימים רק מגרסה 6 של java.

- המשתנים בגאווה סקריפט הם דינמיים. אין צורך להכריז מהו סוג המשתנה והסוג נקבע לפי הערך המוצב בתוך המשתנה:

```
var int = 5; // typeof = number
var real = 5.13; // typeof = number
var str = "my Name is Gilad"; // typeof = string
var bool = true; // typeof = boolean
var v1; // typeof = undefined
```

- אם נציב למשתנה ערך מסוג אחר – המשתנה יהפוך דינמית לסוג החדש. אין צורך בהכרזות מקדימות.

```
var x = 5;
x = "words";
```

- משתנה שרק הוכרז ולא הוצב בו ערך הוא מסוג undefined.

כתיבה לדף

- ניתן גם לכתוב לתוך דף HTML באמצעות הפקודה הבאה:

```
<script>  
    document.writeln("Hello world");  
  
    document.write("<h1 style='text-align: center'> Hello world </h1>");  
</script>
```

- שימו לב – ניתן לכתוב לדף גם תגיות HTML ולא רק טקסט.
- אין הבדל משמעותי בין write ל-writeln שכן כפי שהסברנו בעבר HTML מתעלם ממעברי שורה.

קליטת נתונים מהמשתמש prompt ()

- ישנם מספר דרכים לקלוט נתונים מהמשתמש. אחד הדרכים היא באמצעות הפקודה prompt("..."). הפקודה מחזירה ערך string.

```
var a, b, c;  
a = prompt("Enter first number");  
b = prompt("Enter second number");  
c = a + b;  
document.write(c);
```

- כדי לקלוט מספרים עלינו להוסיף פקודה ההופכת string למספר parseInt().

```
var a, b;  
a = parseInt(prompt("Enter first number"));  
b = parseInt(prompt("Enter second number"));  
document.write("a + b =" + (a + b));
```

• תנאי if-else ותנאי switch-case:

```
switch(expression) {  
  case x:  
    // code block  
    break;  
  case y:  
    // code block  
    break;  
  default:  
    // code block  
}
```

```
if (condition1) {  
  // code to execute if condition1 is true;  
} else if (condition2) {  
  // code to execute if condition1 is false and condition2 is true;  
} else {  
  // code to execute if condition1 and condition2 are false;  
}
```

- התנאים (condition) יכולים לכלול את האופרטורים הבאים:

JavaScript Comparison Operators

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

JavaScript Logical Operators

Operator	Description
&&	logical and
	logical or
!	logical not

- בגאווה סקריפט קיימות כל הלולאות המקובלות בשפות תכנות, נביא כדוגמה שתיים מהן:

```
alert("start");  
for (var i = 1; i < 10; i++) {  
    document.write(i);  
}
```

- לולאת for

```
alert("start");  
var i = 1;  
while (i < 10) {  
    document.write(i);  
    i++;  
}
```

- לולאת while

```
function funcName() {  
    // Code to excute  
}
```

• פונקציה פשוטה (ללא פרמטרים וללא החזרת ערך):

```
function funcName(p1, p2) {  
    // Code to excute  
}
```

• פונקציה עם פרמטרים:

```
function funcName(p1, p2) {  
    // Code to excute  
    return result  
}
```

• פונקציה עם פרמטרים המחזירה ערך:

- הערה comment כותבים בשתי דרכים:

```
/*  
    multi line comment  
*/  
  
alert("hello world"); // comment
```

- ניתן לכתוב או לבטל הערה גם באמצעות הכפתור המיוחד של Visual Studio.
- ניתן לעטוף קוד בהערה על מנת למנוע את הרצתו לצורך בדיקה debugging.

```
/*alert("hello world");*/
```


אוביקטים ומחלקות

אובייקטים ללא מחלקות

- הגדרת אובייקט ללא צורך במחלקה.

```
var person = {  
  firstName: "Gilad",  
  lastName: "Markman",  
  year: 1968,  
  city: "Rehovot",  
  student: true,  
  Age: function () {  
    return 2020 - this.year;  
  }  
};
```

```
alert(person.name);  
person.name = prompt("Enter New Name");
```

```
function Person(first, last) {  
    this.firstName = first;  
    this.lastName = last;  
    this.year = 1900;  
    this.city = "";  
    this.student = null;  
    this.Age = function () {  
        return 2021 - this.year;  
    }  
}  
  
var student1 = new Person("Noa", "Levy");  
var student2 = new Person();  
student2.firstName = "Gali";  
student2.lastName = "Ron";
```

- יצירת אובייקט חדש על בסיס תבנית של אובייקט.

- הפונקציה היא הבנאי של האובייקט.

```
alert("start");  
class Person {  
    constructor(first, last) {  
        this.firstName = first;           // public  
        this.lastName = last;  
        this.year = 1900;  
        this.city = "";  
        this.student = null;  
    }  
  
    Age () {  
        return 2021 - this.year;  
    }  
}  
  
var student = new Person("Noa", "Levy");  
alert(student.firstName);
```

• החל מ- 2015 ניתן להגדיר מחלקות.

• לא מדובר בתכנות מונחה עצמים.

• אין כימוס ואין הורשה.

גאווה סקריפט ו-HTML אירועים

- קוד JS מורץ מייד כשהדף עולה. אנחנו רוצים שהקוד יופעל (יזומן) רק בהתקיים אירוע מסויים: המשתמש לחץ על כפתור, המשתמש שלח טופס לשרת וכד'.
- לצורך כך אנחנו נעזרים בפונקציות ואירועים:
 - הפונקציה מופעלת רק כאשר מזמנים אותה.
 - הזימון נעשה בעקבות אירוע בדף, ובדרך זו אנחנו יוצרים דף HTML דינמי.
- דוגמאות לאירועים:
 - לחיצה על העכבר על תגית מסוימת - onclick
 - לחיצה כפולה על תגית - ondblclick.
 - סיום העלאת הדף - onload.
 - שינוי בשדה input – onchange
 - לחיצה על המקלדת בתיבת טקסט – onkeydown
 - ועוד ...

קריאה לפונקציה בהתקיים אירוע

- על מנת להשתמש באירוע אנחנו מבצעים את הפעולות הבאות:
- מגדירים את הארוע בדף HTML. הארוע שייך לתגית מסויימת בדף.
- כותבים פונקציה שתופעל בעת שהתרחש האירוע
- דוגמה, לחיצה על כפתור:

```
<body>
  <form id="form1">
    <input type="button" id="btn" name="btn" value="click" onclick="clickFunc()"/>
  </form>
</body>
```

```
function clickFunc() {
  alert("onclick");
}
```

- הפונקציה שתופעל בעת הלחיצה:

- זימון פונקציה – באמצעות אירוע (event).

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

The list is much longer: [W3Schools JavaScript Reference HTML DOM Events](https://www.w3schools.com/jsref/dom_obj_event.asp).

- https://www.w3schools.com/jsref/dom_obj_event.asp

- שימוש באירוע onchange של תיבת טקסט.

- דף HTML:

```
<!-- text -->  
<label for="phone">phone:</label>  
<input type="text" id="phone" name="phone" onchange="phoneValidate()" />
```

```
function phoneValidate() {  
    alert("event was fired");  
}
```

- הפונקציה:

- `onblur` – האלמנט מאבד את הפוקוס
- `onfocus` – האלמנט מקבל את הפוקוס
- `onchange` – נעשה שינוי באלמנט
- `oninput` – המשתמש הכניס נתונים לאלמנט
- `onkeydown` – המשתמש לחץ על מקש במקלדת
- `onkeyup` – המשתמש שחרר את המקש במקלדת
- `onkeypress` – המשתמש לחץ על מקש

- בעת קריאה לפונקציה לאחר אירוע (event) ניתן להעביר את האירוע כפרמטר לפונקציה לקבלת מידע נוסף. לדוגמה קבלת המקש שנלחץ ב onkeypress:

```
<!--event parameters-->  
<label for="phone">event parameters example</label>  
<input type="text" id="text" name="text" onkeypress="keyPressFun(event)" />
```

```
function keyPressFun(event) {  
    alert("key: " + event.key + ", keyCode:" + event.keyCode);  
    if (event.key <= "0" || event.key >= "9") {  
        alert("not digit");  
        //return false;  
    } else  
        alert("digit");  
    //return true;  
}
```

- ניתן לבטל את האירוע באמצעות JS ולהחזיר את המצב לפני האירוע, באמצעות החזרת false מהפונקציה.
- לדוגמא, ביטול הקשת התו על ידי המשתמש אם אינו מספר:

```
<input type="text" id="text" name="text" onkeypress="return keyPressFun(event)" />
```

```
function keyPressFun(event) {  
    alert("key: " + event.key + ", keyCode:" + event.keyCode);  
    if (event.key <= "0" || event.key >= "9") {  
        alert("not digit");  
        return false;  
    } else  
        alert("digit");  
    return true;  
}
```

אינטראקציה עם דף HTML

- באמצעות JavaScript אנחנו יכולים ליצור אינטראקציה עם דף HTML. לקרוא מה כתוב בדף ולשנות את האובייקטים בדף.
- ציינו שלושה סוגים של משתנים ב JavaScript:
 - מספר number
 - מחרוזת string
 - בולאני boolean.
- משתנה בגאווה סקריפט יכול להיות גם מסוג אובייקט של HTML. כלומר, משתנה יכול להיות מצביע ל"תגית" או אלמנט של HTML שכתבנו בדף.

Document.getElementById ()

- על מנת לקרוא או לשנות אובייקט (תגית) בדף HTML אנחנו משתמשים בפקודה הבאה:

```
var x = document.getElementById("tagId");
```
- הפקודה מוצאת את האלמנט בדף עם ה-id המבוקש, ומציבה אותו בתוך משתנה x.
- לאחר מכן אנחנו יכולים לקרוא את המאפיינים של האלמנט או לשנות אותם.

- כתיבה למקום מסויים בסף נעשה באמצעות עדכון המאפיין :innerHTML

```
<body>
  <button type="button" id="btn" onclick="btnChange()">change Div</button>
  <br /> <br />
  <div style = "background-color:lightblue; width:300px; height:200px" id="myDiv">
  </div>
</body>
```

```
function btnChange() {
  var div = document.getElementById("myDiv");
  div.innerHTML = "<p> New content </p>";
}
```

```
document.getElementById("myDiv").innerHTML = "<p> New content </p>";
```

- קוד JavaScript:

- בקיצור ניתן לכתוב:

- באמצעות JS ניתן לשנות גם עיצוב CSS כך:

```
function changeStyle() {  
    var div = document.getElementById("myDiv");  
    div.style.color = "red";  
    div.style.backgroundColor = "blue";  
    div.style.border = "solid 3px black";  
    div.style.verticalAlign = "center";  
}
```

- https://www.w3schools.com/jsref/dom_obj_style.asp

דוגמה נוספת – toggle color

- שינוי צבע של אובייקט בהתאם למצבו.

```
function toggleColor() {  
    var div = document.getElementById("myDiv");  
    if (div.style.color == "red") {  
        div.style.color = "blue";  
        div.style.backgroundColor = "red";  
    }  
    else {  
        div.style.color = "red";  
        div.style.backgroundColor = "blue";  
    }  
}
```

```
<!-- select -->  
<label for="colors">Choose a color:</label>  
<select id="colors" name="colors" onchange="selectFun()">  
  <option value="null" selected="selected"></option>  
  <option value="red">Red</option>  
  <option value="blue">Blue</option>  
  <option value="green">Green</option>  
  <option value="yellow">Yellow</option>  
</select>  
<br /> <br />  
<div id="div2"></div>
```

```
function selectFun() {  
  var select = document.getElementById("colors");  
  var div = document.getElementById("div2");  
  var myDiv = document.getElementById("myDiv");  
  
  var i = select.selectedIndex;  
  var color = select.options[i].value;  
  div.innerHTML = "select index= " + i;  
  div.innerHTML = div.innerHTML + " item = " + color;  
  myDiv.style.backgroundColor = color;  
}
```

- דוגמה לקריאה של תיבת select.
- שינוי הצבע בתיבה ישנה צבע בדף.

מחרוזות – בדיקת טופס

Regular Expression

```
var ch1 = str[i];  
var ch2 = str.charAt(i);
```

- ניתן לפנות לתו במחרוזת בשתי דרכים:
- באמצעות הפונקציה `.charAt()`.
- באמצעות האינדקס וסוגריים מרובעים `.str[0]`.

• רשימת פעולות על מחרוזות:

- https://www.w3schools.com/js/js_string_methods.asp
- https://www.w3schools.com/jsref/jsref_obj_string.asp

פונקציות מחרוזת נבחרות

	<code>s1.length();</code>	החזרת אורך המחרוזת
תו ראשון נמצא במקום 0	<code>s1.charAt(k);</code>	החזרת תו במקום k במחרוזת
ch – תו str – מחרוזת מיקום ראשון = 0	<code>s1.indexOf(ch)</code> <code>s1.indexOf(str)</code>	החזרת המיקום הראשון של תו/תחילת תת-מחרוזת מתחילת המחרוזת. (-1) אם לא קיים.
ch – תו str – מחרוזת מיקום ראשון = 0	<code>s1.indexOf(ch, k)</code> <code>s1.indexOf(str, k)</code>	החזרת המיקום הראשון של תו/תחילת תת-מחרוזת ממיקום k במחרוזת. (-1) אם לא קיים.
ch – תו str – מחרוזת מיקום ראשון = 0	<code>s1.lastIndexOf(ch)</code> <code>s1.lastIndexOf(str)</code>	החזרת המיקום האחרון של תו/תחילת תת-מחרוזת מסוף המחרוזת. (-1) אם לא קיים.
ch – תו str – מחרוזת מיקום ראשון = 0	<code>s1.lastIndexOf(ch, k)</code> <code>s1.lastIndexOf(str, k)</code>	החזרת המיקום האחרון של תו/תחילת תת-מחרוזת ממיקום k מסוף המחרוזת במחרוזת. (-1) אם לא קיים.
	<code>s1 = s1.toLowerCase();</code>	מחליף את כל אותיות המחרוזת לאותיות קטנות
	<code>s1 = s1.toUpperCase();</code>	מחליף את כל אותיות המחרוזת לאותיות גדולות

- Regular Expression – היא שיטה לבדיקה אם תבנית מסויימת של תווים מתקיימת במחרוזת נתונה.
- הדרך לשימוש ב RegExp היא כך:
 - מגדירים אובייקט מסוג RegExp הכולל תבנית מסויימת של תווים.
 - באמצעות פעולות test() או match() אנו מבצעים בדיקה אם התבנית שהוגדרה מתקיימת במחרוזת הנתונה.
- קיימים כללים לבניית התבנית באמצעות תווים מוסכמים. לפירוט הכללים ראו:
 - <https://www.w3resource.com/javascript/form/email-validation.php>
 - https://www.w3schools.com/jsref/jsref_obj_regexp.asp

- לצורך הדגמת השיטה נבנה את הטופס הבא:

```
<form id="form1">
  <label for="text">text:</label>
  <input type="text" name="txt" id="txt"/>
  <br /> <br />
  <div id="msg" style="width:400px; height:100px; border:solid"></div>
  <br /> <br />
  <input type="button" value="check" name="btn" id="btn" onclick="chkText()"/>
</form>
```

text:

check

- לחיצה על כפתור check תריץ את הפונקציה.


```
function chkText() {  
    var text = document.getElementById("txt").value;  
    var msg = document.getElementById("msg");  
    var reg = /abc/;  
    if (reg.test(text))  
        msg.innerHTML = "True";  
    else  
        msg.innerHTML = "False";  
}
```

```
var reg = /^abc/;
```

```
var reg = /abc$/;
```

- בדיקה אם במחרוזת קיים רצף של התווים "abc" במקום כלשהו.
- תבנית regExp מוגדרת באמצעות ./.../

- בדיקה אם המחרוזת מתחילה ב abc
- בדיקה אם המחרוזת מסתיימת ב abc

```
var reg = /[2a8]/;
```

```
reg = /\d/; reg = /[0-9]/
```

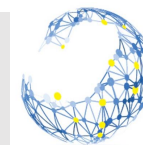
```
reg = /[a-z]/i; reg = /[a-zA-Z]/
```

```
reg = /\w/; reg = /[a-zA-Z0-9]/;
```

```
reg = /(Gilad|Markman)/;
```

- סוגריים מרובעים – האם המחרוזת מכילה לפחות אחד מהתווים.
- האם המחרוזת מכילה ספרות.
- המחרוזת מכילה אותיות גדולות או קטנות.
- case-insensitive matching - i
- המחרוזת מכילה אותיות או ספרות.
- המחרוזת מכילה Gilad או Markman

Regular Expression



```
reg = /^0\d{9}$/;
```

```
reg = /^[a-zA-Z]+@[a-zA-Z]{3,4}$/;
```

```
reg = /\W/;
```

```
reg = /\s/;
```

```
reg = /\D/;
```

- מחרוזת מתחילה בספרה 0 וכוללת לאחריה 9 ספרות.
- המחרוזת מתחילה באות אנגלית אחת או יותר (+), לאחר מכן מכילה תו @ אחד, ולאחריו 3-4 אותיות באנגלית.
- המחרוזת כוללת תווים מיוחדים (נכריח להוסיף תו מיוחד בסיסמה).
- המחרוזת כוללת רווח (למנוע רווחים במחרוזת).
- המחרוזת כוללת תווים שאינם ספרות (לבדוק שהוכנסו רק ספרות)

דוגמה אימות טופס

- בניית טופס באמצעות טבלה עם מקום להערות.

User Name	<input type="text"/>	You must enter user name
Password	...	password must contain at least 7 charactes
confirm password	confirm password doesn't mutch password
e-mail	gilad@markman.co.	Invalid email
Tell	052-945878	phone number is illegal
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other	
תחביבים	Footbal <input type="text"/>	
תאריך לידה	mm/dd/yyyy <input type="text"/>	
שאלת אימות	שם הכלב הראשון שלך <input type="text"/>	<input type="text"/>
	<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

בניית פונקציה ראשית onSubmit

```
<form id="form1" onSubmit="return validateForm()">
```

- הוספת אירוע onSubmit לטופס:

- הוספת אירוע onclick לכפתור submit:

```
<input type="submit" name="submit" onclick="return validateForm()" />
```

```
function validateForm() {  
  
    var res = true;  
    res = userNameVal() && res;  
    res = passwordVal() && res;  
    res = confirmFun() && res;  
    res = confirmemail() && res;  
    res = confirmPhone() && res;  
    return res;  
}
```

- בניית הפונקציה הראשית לבדיקת הטופס:
- ביטול האירוע במקרה והפונקציה מחזירה false.
- צבירה של תנאים בוליאניים באמצעות "וגם".
- סדר האופרנדים (הפונקציה מבוצעת קודם ל-"וגם").

```
function userNameVal() {  
    var userName = document.getElementById("userName").value;  
    var msgBox = document.getElementById("userNameMsg");  
    if (userName.length == 0) {  
        msgBox.innerHTML = "You must enter user name";  
        return false;  
    }  
    if (!isLetter(userName[0])) {  
        msgBox.innerHTML = "User name must start with a letter ";  
        return false;  
    }  
    msgBox.innerHTML = "";  
    return true;  
}
```

- בדיקת אם הערך ריק.

- בדיקת אות התחלתית.

- כתיבת הערות ומחיקתם

```
function passwordVal() {  
    var pass = document.getElementById("password").value;  
    var msgBox = document.getElementById("passwordMsg");  
    if (pass.length < 7) {  
        msgBox.innerHTML = "password must contain at least 7 charactes";  
        return false;  
    }  
    var specialChar = /[!#$%^&*()-+]/; //regular Expretion  
    if (!specialChar.test(pass)) {  
        msgBox.innerHTML = "password must contain one special char";  
        return false;  
    }  
    msgBox.innerHTML = "";  
    return true;  
}
```

- מספר תווים מינימלי
- קיום תו מיוחד.

- הבדיקה תוך כדי הזנה (אירוע onchange) וגם בפונקציה הראשית.

```
<input type="password" name="confirm" id="confirm" onchange="confirmFun()" />
```

```
function confirmFun() {  
    var pass = document.getElementById("password").value;  
    var confirm = document.getElementById("confirm").value;  
    var msgBox = document.getElementById("confirmMsg");  
  
    if (pass !== confirm) {  
        msgBox.innerHTML = "confirm password doesn't mutch password";  
        return false;  
    }  
    msgBox.innerHTML = "";  
    return true;  
}
```

- הפונקציה לבדיקה:

- בדיקה בעת אירוע oninput וכן בפונקציה הראשית:

```
<input type="text" name="phone" id="phone" oninput="confirmPhone()"/>
```

```
function confirmPhone() {  
    var phone = document.getElementById("phone").value;  
    var msgBox = document.getElementById("phoneMsg");  
  
    var reg = /^0{1}(2|3|4|6|8|9|5[2-8]|73)-[1-9]\d{6}$/;  
    if (!reg.test(phone)) {  
        msgBox.innerHTML = "phone number is illegal";  
        return false;  
    }  
    msgBox.innerHTML = "";  
    return true;  
}
```

- הפונקציה- שימוש ב regExp:

```
function confirmemail() {  
  
    var email = document.getElementById("email").value;  
    var msgBox = document.getElementById("emailMsg");  
  
    reg = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,4}+$/;  
    if (!reg.test(email)) {  
        msgBox.innerHTML = "Invalid email";  
        return false;  
    }  
  
    msgBox.innerHTML = "";  
    return true;  
}
```

שימוש ב Regular Expression.

.RegExp

- * • 0 או יותר פעמים
- + • 1 פעמים או יותר
- \\w • אות או ספרה
- ? • 0 או 1 פעמים
- \\. • התו נקודה "."
- {2,4} • 2 – 4 פעמים.

Debugging DevTools – F12

- מציאת טעויות ("באגים") בקוד היא פעולה סיזיפית ומתישה. לצורך כך קיימים בדפדפנים כלי רב עוצמה העוזר לנו לבצע תיקון טעויות (debugging).
- אנחנו נדגים את הכלי של ה chrom, אולם בדפדפנים האחרים הכלים דומים.
- הכניסה לכלי הפיתוח נעשה מתוך הכרום על ידי לחיצה על F12.



HTML קוד

מסך הפיתוח עיצוב

עיצוב CSS



The screenshot shows the Chrome DevTools interface. The 'Elements' panel on the left displays the HTML structure of a registration form, with the 'td' element containing 'User Name' selected. The 'Styles' panel on the right shows the CSS rules for the selected element, including a custom class with a border and padding, and the user agent default styles. A red circle highlights the 'Styles' panel.

```

<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <div>...</div>
    <ul>...</ul>
    <br>
    <br>
    <form id="form1" onsubmit="return validateForm()">
      <table>
        <tbody>
          <tr>
            <td style="width:30%">User Name</td>
            <td style="width:30%">...</td>
            <td id="userNameMsg" class="valMsg" style="width:40%"></td>
          </tr>
          <tr>...</tr>
          <tr>...</tr>
          <tr>
            <td>e-mail</td> == $0
          </tr>
        </tbody>
      </table>
    </form>
  </body>
</html>
  
```

```

border {
  border-collapse: collapse;
  width: 60%;
}
table user agent stylesheet {
  border-collapse: separate;
  border-spacing: 2px;
}
  
```

border: 1.200px solid #ccc; padding: 10px; width: 60%;

border: 1.200px solid #ccc; padding: 10px; width: 60%;

padding: 10px; width: 60%;

border: 1.200px solid #ccc; padding: 10px; width: 60%;

מצביע לדף

מסך הפיתוח קוד

מעבר לקוד

The screenshot shows the Chrome DevTools interface. The 'Sources' panel is active, displaying the file structure on the left and the source code of 'Registration.html' in the center. The code includes a meta charset declaration, a title in Hebrew, and a form with a text input field for 'user Name' and a validation message. The 'Console' panel is visible at the bottom, and the 'Elements' panel is partially visible on the left.

קבצים של הדף

ה console

javaScript Debugging

The screenshot shows the Chrome DevTools interface with a breakpoint set on line 4 of the `validateForm` function. The code is as follows:

```

1 function validateForm() {
2
3
4   var res = true;
5   res = userNameVal() && res;
6   res = passwordVal() && res;
7   res = confirmFun() && res;
8   res = confirmemail() && res;
9   res = confirmPhone() && res;
10  return res;
11 }
12
13 function userNameVal() {
14   var userName = document.getElementById("userName").value;
15   var msgBox = document.getElementById("userNameMsg");
16   if (userName.length == 0) {
17     msgBox.innerHTML = "You must enter user name";
18     return false;
19   }
20   if (!isLetter(userName[0])) {
21     msgBox.innerHTML = "User name must start with a letter ";
22     return false;
23   }
24   msgBox.innerHTML = "";
25   return true;
26 }
27
28 function passwordVal() {

```

The DevTools interface shows the following components:

- Breakpoint:** A red dot on line 4 of `Registration.js` is circled in red. A red box labeled "Break point" points to it.
- Debugger:** The top right of the editor shows the debugger toolbar with a red circle around the "Step over next function call" button. A red box labeled "התקדמות צעד אחד צעד בקוד" (Advance one step in the code) points to this button.
- Call Stack:** Shows the current call stack with `validateForm` at the top.
- Scope:** Shows the local scope with `res: undefined`.
- Breakpoints:** A list of breakpoints with `Registration.js:4 var res = true;` checked.

Break point

התקדמות צעד אחד צעד בקוד

בדיקת תוכן משתנים

The screenshot shows the Chrome DevTools interface with the Sources panel open to Registration.js. A red circle highlights the line where the variable `userName` is assigned the value `"Gilad"`. The right-hand sidebar shows the 'Debugger paused' state with the 'Scope' panel displaying `userName: "Gilad"`. The bottom console panel shows the value of `userNameVal` as `"Gilad"`.

הצבעה על משתנה תראה לנו את התוכן שלו

הדפסה של המשתנה תדפיס ב console את תוכנו

- הקדמה – מיקום כתיבת הקוד.
- פקודות בסיסיות ופונקציות.
- אובייקטים ומחלקות.
- אירועים Events.
- איטראקציה עם דף HTML – `document.getElementById("id")`
- מחרוזות RegExp ואימות טפסים.
- DevTools - F12
- הסבר על המטלה