

בדיקת תקינות הטופס

לרוב הטפסים שנבנה יש לבצע בדיקות תקינות לפני המשלוח לשרת. מטרת הבדיקות לוודא כי המשתמש מלא את הנתונים הנדרשים, וכך שהנתונים כתובים בפורמט נכון.

לדוגמה:

- אסור ששדות חובה יישארו ריקים.
- אורך סיסמה יהיה לפי כללים שנקבעו באתר.
- כתובת דוא"ל חוקית.
- שנת לידה תכיל מספרים בלבד.

בדיקות התקינות נעשות בצד הלקוח, לאחר לחיצה על כפתור **שלח**. אם נמצאו שגיאות יש להציג הודעה מתאימה ולחזור לטופס לתיקון השגיאות.

לאחר תיקון כל השגיאות מועבר הטופס לשרת.

כדי שהמשתמש ידע מה לתקן, יש להציג לו הודעה מתאימה.

את ההודעה נוכל להציג:

בדרך א' - באמצעות תיבת alert :

או - **בדרך ב'** -

הצגת הודעת שגיאה מתאימה ליד השדה הלא תקין, הודעה שתיעלם כשהמשתמש יתקן את הטעון תיקון וישלח שוב את הטופס:

שם משפחה קצר מדי או לא קיים	<input type="text"/>	שם משפחה:
שם פרטי קצר מדי או לא קיים	<input type="text"/>	שם פרטי:
כתובת דוא"ל קצרה מדי או לא קיימת	<input type="text"/>	דוא"ל:
<input type="button" value="שלח"/>		

נבחר בדרך ב'.

בדיקות התקינות נעשות בשפת javascript בתגית <script>
 השפה רגישה לגודל אות.
 שמות הפעולות בשפת javascript מתחילות תמיד באות קטנה !!!

הגדרת השדה להודעת השליאה:

כדי להציג את ההודעה יש להוסיף לטופס, ליד כל שדה קלט, שדה טקסט נוסף שיהיה במצב מוסתר ויוצג רק כאשר מתגלה שגיאת הקלדה של המשתמש.

```
<input type = "text" id = "mFName" size = "30"
style = "display: none; background-color: Silver; font-weight:bold;"
disabled = "disabled" />
```

<code>input type = "text"</code>	שדה קלט מסוג טקסט
<code>id = "mfName"</code>	שם השדה יוגדר באמצעות id בלבד (אין צורך ב- name כי הוא לא מגיע לשרת). שם השדה יכיל את שם שדה הטופס אליו הוא משויך, מכיוון ששדה זה יכיל הודעת שגיאה, נוסף לו את האות m עבור message (הודעה)
<code>style = "display: none;"</code>	נמצא במצב מוסתר
<code>background-color: Silver; font-weight: bold;"</code>	צבע הרקע אפור-כסף והכתב מודגש
<code>disabled = "disabled"</code>	במצב disabled השדה אינו פעיל, אין אפשרות גישה אליו

דוגמה לשורה בטופס:

שים לב למשפט ה- return כשיתרחש האירוע submit:

```
<form name = "frm1" method = "post" runat = "server"
onsubmit = "return checkForm();" >
```

```
<table border = "1" align = "center" >
<tr>
```

השדות הגלויים בשורה בטופס: כותרת השורה ושדה הקלט:

```
<td> פרטי שם </td>
<td><input type = "text" name = "fName" id = "fName"></td>
```

השדה המוסתר בשורה: עמודה נוספת בשורה, העמודה שתכיל את הודעת השגיאה:

```
<td>
<input type = "text" id = "mFName" size = "30"
style = "display: none; background-color: Silver;
font-weight: bold;"
disabled = "disabled" />
</td>
</tr>
```

פונקציית הבדיקה

נוסיף ב- head פונקציה לבדיקת תקינות הטופס. שם הפונקציה יהיה זהה לשם עליו הצהרנו ב- onsubmit. לכל שדה בטופס נכתוב הוראות בדיקה מתאימות. אם התגלתה טעות בשדה (שדה ריק, שדה מספרי אינו מכיל רק ספרות, כתובת דוא"ל אינה תקינה, שדה אימות סיסמה אינו תואם את שדה הסיסמה וכד') נצמיד לשדה הערת שגיאה מתאימה ונגרום לשדה להיות גלוי ופעיל.

למשל: **בדיקה שהמשתמש הכניס שם פרטי:**

```
//--- בדיקת תקינות הטופס ---
function CheckForm()
{
    //--- שם פרטי קצר מ-2 תווים ---
    var fName = document.getElementById("fName").value;
    if (fName.length < 2)
    {
        document.getElementById("mfName").value = "שם פרטי קצר מדי או לא קיים";
        document.getElementById("mfName").style.display = "inline";
        return false;
    }
    else
        document.getElementById("mfName").style.display = "none";
}
```

<code>document.getElementById("שם השדה")</code>	הפעולה מחזירה הפניה לאובייקט שזהו ה-id שלו. לאובייקט יש מאפיינים אליהם ניתן לפנות:
<code>value</code>	המאפיין/תכונה של האובייקט: תוכן האובייקט.
<code>var fName = document.getElementById("fName").value;</code>	השמת ערך/תוכן התכונה במשתנה. (שים לב: אין קשר פיסי בין שם התכונה ושם המשתנה. הקשר הוא קשר לוגי בלבד, עבור המתכנת, ומסייע להתאים בין המשתנים של הפעולה לערך השדות המתאימים.)
<code>document.getElementById("mfName").value = "מחרוזת טקסט";</code>	השמת ערך בתכונה. שים בשדה mfName את מחרוזת הטקסט הנתונה.
<code>style.display = "inline"</code>	הפיכת השדה לפעיל

כל בדיקות התקינות מתבצעות בתוך הפונקציה `checkForm()` שמוגדרת ב- `head`. משפט הזימון לפעולה נכתב בתגית `form` ומופעלת בעת לחיצה על כפתור `submit`:

```
<form name = "frm1" method = "post" runat = "server" onsubmit = "return checkForm();" >
```

הפעולה מחזירה ערך בוליאני **אמת** - `true` או **שקר** - `false` (שים לב למשפט הזימון: בלחיצה על `submit` **החזר** ערך מהפעולה) ולכן, בדיקת התקינות מבצעת:

אם ערך שדה הקלט אינו תקין,

א. שים בשדה ההודעה של שדה זה הודעת שגיאה מתאימה.

ב. הפוך את השדה לפעיל.

ג. החזר `false`

אחרת - הפוך את השדה ללא פעיל

אם הערך שדה כשלהו אינו תקין, תציב הפעולה הודעת שגיאה מתאימה ותחזיר ערך false. הטופס יוצג כשלצד השדה השגוי תופיע הודעת שגיאה.

```
var fName = document.getElementById("fName").value;
if (fName.length < 2)
{
    document.getElementById("m1Name").value = "שם משפחה קצר מדי או לא קיים";
    document.getElementById("m1Name").style.display = "inline";
    return false;
}
```

טופס קבלת פרטים

שם משפחה קצר מדי או לא קיים	<input type="text"/>	שם משפחה:
	<input type="text"/>	שם פרטי:
	<input type="text"/>	דוא"ל:
	<input type="button" value="שלח"/>	

לאחר תיקון השגיאה ולחיצה חוזרת על "שלח", ייבדק ערך השדה. אם עכשיו הוא תקין, יחזור שדה השגיאה של שדה זה להיות לא פעיל ומוסתר. (display = "none")

```
else
    document.getElementById("m1Name").style.display = "none";
```

טופס קבלת פרטים

שם משפחה קצר מדי או לא קיים	<input type="text"/>	שם משפחה:
	<input type="text"/>	שם פרטי:
	<input type="text"/>	דוא"ל:
	<input type="button" value="שלח"/>	

תרגיל:

הוסף לטופס שיצרת בדיקות תקינות לכל שדה. אם לא הוקלד מידע בשדה או שהמידע קצר מדי (פחות מ- 2 תווים לשם, פחות מ- 6 תווים לכתובת דוא"ל) תוצג הודעה מתאימה.

שים ♥ : הקפד לנסח את הודעות השגיאה כך שהמשתמש יבין את מהות השגיאה וידע לתקנה.

בדיקות תקינות fצה דוא"ל:

aa@a.a	אורך מינימלי - 6 תווים: יכיל בדיוק סימן @ אחד. מיקום ה-@ לא יהיה לפני תו שלישי ולא תו אחרון. יכיל נקודה אחת לפחות, והיא תהיה במרחק של לפחות 2 תווים אחרי ה-@. הנקודה לא תהיה התו הראשון או האחרון בכתובת הדוא"ל. לא יכיל אותיות וסימנים אסורים.
--------	--

הכתובת אותה הזין המשתמש הינה מחרוזת. כדי לבצע את בדיקות התקינות הדרושות עלינו להכיר פונקציות לטיפול במחרוזות בשפת javascript.

אורך מחרוזת:

התכונה אורך המחרוזת סופרת ומחזירה את מספר התווים במחרוזת. `str.length`

מחרוזת ריקה (שאינ בה תווים) אורכה 0.

במחרוזת לא ריקה נספרים התווים החל מהתו הראשון שמקומו 0 ועד התו האחרון שמקומו `str.length - 1`

למשל: המחרוזת "abcd" אורכה 4 תווים:

	0	1	2	3
str	a	b	c	d

בדיקות הקשורות באורך המחרוזת:

- האם מחרוזת ריקה?
`if (str.length == 0) ...`
או
`if (str == "") ...`
- האם אורך המחרוזת קצר מ-6 תווים?
`if (str.length < 6) ...`

מציאת תו במחרוזת:

פונקציה המחזירה את המיקום הראשון של התו במחרוזת. `str.indexOf('@')`

אם התו נמצא: יוחזר מיקומו מספר בין 0 (מקום ראשון) וה-`length` שלה (לא כולל).
אם התו לא נמצא: יוחזר המספר -1

פונקציה המחזירה את מקומו של התו האחרון במחרוזת: `str.lastIndexOf('@')`

פונקציה המחזירה את המיקום הבא של התו אחרי התו ששמור במשתנה שבפרמטר השני:

```
var atSign = str.indexOf('@');
var dotSign = str.indexOf('.', atSign);
```

התו באינדקס נתון:

פונקציה המחזירה את התו שנמצא באינדקס `k` במחרוזת. דוגמה עבור `k = 2`:
`var str = "abcde";`
ערכו של `k` הוא: $0 \leq k < \text{str.length}$
`'c' ← str.charAt(2);`

פונקציות ב-javascript נכתבות באות קטנה. **שים ♥:**
כל מילה החל במילה השנייה מתחילה באות גדולה

בדיקות תקינות לכתובת דוא"ל:

שים ♥: לכל סוג שגיאה יש לשים הודעת שגיאה מתאימה בשדה meMail

ההוראה ב-javascript	בדיקת התקינות
<code>if (eMail.length < 6) ...</code>	אורך מינימלי - 6 תווים: aa@a.a
<code>var atSign = eMail.indexOf('@'); if (atSign != eMail.lastIndexOf('@')) ...</code>	יכול בדיוק סימן @ אחד. אם מיקומו של סימן ה-@ הראשון שונה ממיקומו של סימן ה-@ האחרון
<code>var size = eMail.length; if (atSign < 2 atSign == size-1) ...</code>	מיקום ה-@ לא יהיה לפני תו שלישי ולא תו אחרון. size שומר את אורך מחרוזת הדוא"ל. atSign שומר את מיקומו של תו ה-@ התו השלישי במחרוזת הוא התו במקום 2 במחרוזת בעלת 10 תווים יהיו כל התווים במקומות 0 עד 0, שה"כ 10 מקומות. אם מיקומו של @ קטן מ-2 או מיקומו שווה למיקום התו האחרון במחרוזת...
<code>var dotSign = eMail.indexOf('.', atSign); if (dotSign - atSign < 2) ...</code>	יכול לפחות נקודה אחת, והיא תהיה במרחק של לפחות 2 תווים אחרי ה-@.
<code>if (dotSign == 0 dotSign == size-1) ...</code>	הנקודה לא תהיה התו הראשון או האחרון בכתובת הדוא"ל.

מהות הבדיקה:

אם לא מתקיים הכלל בבדיקת התקינות, תוצג הודעת שגיאה.

למשל:

אם דרשנו אורך מינימלי של 6 תווים,

תתבצע הבדיקה: אם האורך קטן מ-6 תווים אז יש להציג הודעת שגיאה.

אם דרשנו שנקודה לא תהיה בהחלה או הסוף הכתובת

תתבצע הבדיקה: אם מיקום הנקודה הוא במיקום הראשון (מקום 0) או במיקום האחרון (מיקום size-1)

אז יש להציג הודעת שגיאה.

שים ♥: בתנאי המורכב מכמה משפטי תנאי פשוט, חובה לרשום את שם המשתנה הנבדק בכל תנאי פשוט.

לדוגמה: בכתיבה מתמטית, כשרוצים לבדוק האם הערך שבמשתנה x הוא בין 3 ל-8 נרשום: $3 < x < 8$
בשפות תכנות נרשום: $x < 8 \ \&\& \ x > 3$ x קטן מ-3 וגם x קטן מ-8

כתובת דוא"ל לא תכיל אותיות וסימנים אסורים:

1. ניצור מחרוזת (בשם badStr) שתכיל את כל הסימנים האסורים.
2. נבדוק בלולאה את כל תווי הכתובת. אם אחד התווים נמצא במחרוזת badStr אז הכתובת לא תקינה (כי היא מכילה תווים אסורים).

מבנה הלולאה:

<pre>var i = 0; while (i < str.length) { str.charAt(i) בדיקת i++; }</pre>	<pre>i ← 0 כל עוד (i קטן מאורך המחרוזת str) { ביצוע בדיקות התקינות הדרושות לתו במקום ה-i i++ (קידום i ב-1) }</pre>
--	--

שים ♥ : סימן " (גרשיים) הוא סימן אסור בכתובת דוא"ל. הסימן מתפקד כתחום מחרוזת. איך נבדוק אם הכתובת מכילה " ? נשמור אותה במשתנה :

```
var quot = '\"';
```

ונבצע עבורו בדיקה נפרדת.

נרשום את הפונקציה האם-המחרוזת-תקינה ? isValidString המקבלת את כתובת הדוא"ל (str) ובודקת את קיומם של התווים האסורים.
אם נמצאו תווים אסורים, תחזיר הפונקציה שקר false (לא תקינה)
ואם לא נמצאו תווים אסורים, תחזיר הפונקציה אמת true (כן תקינה).

--- פעולה המחזירה אמת אם המחרוזת תקינה ושקר אחרת ---

```
function isValidString(str)
{
    //--- אם מכילה גרשיים - לא תקין ---
    var quot = '\"'; // \
    if (str.indexOf(quot) != -1)
        return false;

    //--- אם מכילה את אחד התווים הבאים - לא תקין ---
    var badStr = "$%^&*()_+[]{}<>?עפצקקשת";

    var i = 0, p, ch;
    while (i < badStr.length) {
        ch = badStr.charAt(i);
        p = str.indexOf(ch);
        if (p != -1)
            return false;
        i++;
    }
    return true; //--- הכל תקין ---
}
```

עבור כל אחד מהתווים שבמחרוזת badStr,
חפש את מיקומו במחרוזת str
p ← str (ערכו של p אינו -1), החזר שקר
אם נמצא כזה תו יוחזר מקומו, ואם לא נמצא יוחזר (-1)

כדי שנוכל להציג בכל פעם את הודעת השגיאה המתאימה, ניצור משתנה בשם errorMessage שיכיל מחרוזת ריקה:

```
var errorMessage = "";
```

בכל שלב של הבדיקה, אם התגלתה שגיאה, נשים בו הודעה מתאימה:

```
if (eMail.length < 6)
{
    errorMessage = "כתובת דוא"ל קצרה מדי או לא קיימת";
    valid = false; (*)
}
```

```
var atSign = eMail.indexOf('@');
if (atSign != eMail.lastIndexOf('@'))
{
    errorMessage = "יש יותר מ-@ אחד";
    valid = false; (*)
}
// וכך הלאה לכל בדיקה
```

הפעלת הפונקציה isValidString:

```
if (isValidString(eMail) == false)
{
    errorMessage = "נמצאו תווים אסורים";
    valid = false; (*)
}
```

בסיום נבדוק: אם המחרוזת אינה ריקה - יש לשים במקום של הודעת השגיאה את תוכנו של errorMessage ולהציגו, אחרת - הכל תקין.

```
if (errorMessage != "") (*)
{
    document.getElementById("meMail").value = errorMessage;
    document.getElementById("meMail").style.display = "inline";
    return false;
}
else
    document.getElementById("meMail").style.display = "none";
```

אפשרות נוספת: (*)

```
var valid = true;
ליצור משתנה בשם valid (תקין) שיקבל בהתחלה ערך אמת true.
אם נמצאה שגיאה, נשנה את ערכו של valid לשקר false
```

```
if (valid == false) {...}
בסיום נבדוק אם ערכו של valid הוא שקר, נציג את הודעת השגיאה:
```