

תכנות באינטרנט

Strings

גלעד מרקמן

קריית החינוך פארק המדע,

נס ציונה



קריית החינוך
השש שנתית
פארק המדע

עמיד של חדשנות



- מחרוזת היא סדרה של תווים, כגון: "Gilad Markman born in 1968".
- למשתנה מסוג מחרוזת יש מאפיינים ופונקציות שמאפשרים לנו לטפל במחרוזות בצורה קלה ומהירה.
- לכל תו במחרוזת יש אינדקס (מיקום) המתחיל ב- 0 (צד שמאל). במחרוזת למעלה התו במיקום 0 הוא 'G'. התו במיקום 5 הוא רווח (space).
- אורך המחרוזת (length) בדוגמה למעלה הוא של 26 תווים.
- השימוש בפונקציות של המשתנה מסוג מחרוזת נעשה באמצעות שם המשתנה ולאחריו נקודה "." ושם המאפיין או הפונקציה, לדוגמא:

```
var str = "Gilad Markman born in 1968";  
var len = str.length;
```

```
var str = "Gilad Markman born in 1968";  
document.write(str + "<br/>");  
document.write(str.length + "<br/>");  
document.write(str.charAt(0) + "<br/>");  
document.write(str.charAt(str.length-1));
```

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

```
Gilad Markman born in 1968  
26  
G  
8
```

- בשני המקרים ניתן להשתמש במשתנה מספרי בתוך הסוגריים, למשל:

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

- ניתן רק לקרוא את התווים ולא לשנות אותם.

- לכל תו קיים קוד המכונה קוד אסקי (Ascii code).

```
var str = "Gilad";  
alert(str.charCodeAt(0));
```

- הפיכת תו למספר קוד:

```
str = String.fromCharCode(120);  
alert(str);
```

- הפיכת קוד אסקי לתו:

- https://www.w3schools.com/charsets/ref_html_ascii.asp

פונקציות מחרוזת נוספות

	<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

```
str = str1 + "Shalom" + str2;
```

```
var str = "Gilad Markman born in 1968";  
var s = str.slice(6, 13);
```

```
str = "    Gilad M    ";  
s = str.trim();  
alert(s.length);
```

- שירשור מחרוזת נעשה באמצעות אופרטור החיבור.
- חיתוך מחרוזת מהמיקום 6-12.
 - התוצאה: Markman
- ביטול רווחים בתחילה ובסוף המחרוזת (לא באמצע).

רשימת פעולות על מחרוזות:

https://www.w3schools.com/js/js_string_methods.asp

https://www.w3schools.com/jsref/jsref_obj_string.asp

- 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

```
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)

- שימוש במחרוזות ותווים ושירור מחרוזת.
- למדנו מה זה קוד אסקי
- למדנו שניתן לקרוא לתווים במחרוזת לפי האינדקס של התו (לא ניתן לעדכן).
- הכרנו פונקציות של מחרוזת, כגון: `indexOf()`; `slice()`; `trim()`
- הדגמנו שימוש ב `regular expression` לבדיקת אימות של מחרוזות.
- הפננו לרשימת הפעולות של האובייקט `String` באינטרנט.
- בשיעור הבא נדגים את השימוש בחומר שנלמד לצורך בדיקת תקינות טופס