

לתרגול אדפטיבי, בדיקת תשובות אוטומטית, ועוזר בינה מלאכותית חינם, היכנסו ל-www.TIRGUL.ai

פעולות קיצור דרך

הדוגמאות על `string s = "Hello";`

פעולה	מה היא מחזירה
<code>s.Contains("ell")</code>	האם הטקסט מופיע בה: <code>true</code>
<code>s.Substring(1, 3)</code>	3 תווים החל ממיקום 1: <code>ell</code>
<code>s.IndexOf('l')</code>	מיקום המופע הראשון: <code>2</code>
<code>s.IndexOf('l', 3)</code>	החיפוש מתחיל ממיקום 3: <code>3</code>
<code>s.LastIndexOf('l')</code>	מיקום המופע האחרון: <code>3</code>

`IndexOf` ו-`LastIndexOf` מקבלות תו או טקסט: `s.IndexOf("lo")` מחזירה 3. הפעולות לא משנות את המחרוזת עצמה, אלא מחזירות ערך חדש. אות גדולה ואות קטנה הן תווים שונים: `s.Contains("h")` היא `false`.

השוואת מחרוזות

משווים שתי מחרוזות באמצעות `==`.

```
string word = Console.ReadLine();
if (word == "yes")
{
    Console.WriteLine("Great");
}
```

שילוב של כמה פעולות

```
string t = "Dan Cohen";
int space = t.IndexOf(' ');
Console.WriteLine(t.Substring(0, space));
Console.WriteLine(t[space + 1]);
```

הפלט:

Dan
C

אורך ותו במיקום

`Length` מחזירה את אורך המחרוזת. סוגריים מרובעים אחרי שם המחרוזת מחזירים את התו שבמיקום. המיקומים מתחילים ב-0, ולכן המיקום האחרון הוא `Length - 1`. גם רווח הוא תו: הוא נספר ותופס מיקום.

```
string s = "Hello";
Console.WriteLine(s.Length);
Console.WriteLine(s[0]);
Console.WriteLine(s[1]);
Console.WriteLine(s[s.Length - 1]);
```

הפלט:

5
H
e
o

מעבר על כל התווים

`s[i]` הוא `char`, ולכן משווים אותו לתו בגרש בודד.

```
int count = 0;
for (int i = 0; i < s.Length; i++)
{
    if (s[i] == 'l')
    {
        count++;
    }
}
Console.WriteLine(count);
```

הפלט:

2

מעבר מהסוף להתחלה:

```
for (int i = s.Length - 1; i >= 0; i--)
```