

22.9.2017

## סיכום של מה שלמדנו עד כה:

לרוב (אבל לא תמיד) תוכנית מחשב בנויה משלושה חלקים: (1 קלט, 2 עיבוד, 3 פלט.

כדי לקבל קלט לתוך התוכנית (לקרוא ממסך הקונסול) משתמשם בפקודה:  
`Console.ReadLine()`

על מנת שנוכל לשמור על הערך של הקלט יש להגדיר משתנים.

משתנה זה תא בזיכרון המחשב (ram) שיש לו סוג ושם. הסוגים שלמדנו עד היום הם: `int` עבור שלמים, `double` עבור מספרים עם נקודה עשרונית, ו- `string` עבור כל התווים האחרים (מחרוזת בעברית)

שם המשתנה מתחיל באות (באנגלית כמובן) ויכול גם להכיל ספרות. נשתדל שלשמות תהיה משמעות של מה שהמשתנה מסמל עבור התוכנית. דוגמאות:

<code>int num1;</code>	זהו משתנה מסוג שלם ששמו הוא <code>num1</code> (ומיועד להכיל מספר כלשהו).
------------------------	--

<code>double celsius;</code>	זהו משתנה שיכול להכיל מספרים עם שבר עשרוני ששמו הוא <code>celsius</code> (ומיועד להכיל טמפרטורה)
------------------------------	--

<code>string name;</code>	זהו משתנה מסוג מחרוזת שנקרא: <code>name</code> (וכנראה יכיל שם)
---------------------------	---

נחזור לנושא הקלט. כדי לקלוט מספר לתוכנית, צריך קודם להגדיר את המשתנה שאליו המספר יכנס, ואז לקרוא אותו מהקונסול:

```
int num1;  
num1 = int.Parse(Console.ReadLine());
```

שימו לב שפקודת ה- `Console.ReadLine()` 'עטופה' בתוך: `int.Parse` - זה הכרחי עבור המחשב כדי לבין שמה שנקיש יהיה מספר שלם. עבור מספר עם שבר נעטוף את הקריאה ב: `double.Parse`, עבור קריאה של מחרוזת לא צריך לעטוף בכלום.

סימן השווה = מיועד להשמת ערך במשתנה. מה שמשמאל לשווה יכנס למה שמשמאלו, למשל:

```
int x;  
x = 10;
```

הגדרנו שטח (תא) בזיכרון שיכול להכיל מספר שלם, שמו היה x. הכנסנו לו את הערך 10.

מה יהיה ערכו של y אחרי סדרת הפקודות הבאה?

```
double y;  
y = 12;  
y = y + 1;
```

אם אמרתם 13.0 אז צדקתם.

מה יהיה ערכו של x אחרי פקודות הבאות:

```
int x = 8;  
int y = 3;  
x = x+y;  
y = y+x;
```

אם אמרתם 14, אז צדקתם. איך זה קורה? קודם x מקבל 8 ו-y מקבל 3. הפקודה:  $x = x+y$ ;

מחברת את 8 ו-3 ודורסת את הערך הקודם של x (8) וכעת יש בו 11.  
הפקודה:  $y = x+y$ ; מחברת את ה-x החדש שהוא 11 ל-y שהוא 3 ומתקבל הערך 14. הערך 14 דורס את הערך הקודם של y ולכן בסוף יהיה ב-y הערך 14.

כדי לבצע השמה של מחרוזת, נשתמש בצמד של גרשיים. לדוגמה:

```
string phrase = "programming is fun";
```

שם המשתנה המחרוזתי הוא phrase.

כפי ששמתם לב השתמשתי בקיצורים. הגדרתי שם משתנה ונתתי לו ערך באותה שורה. כלומר במקום לכתוב:

```
int x;  
x = 8;
```

כתבתי בפשטות:  $\text{int } x = 8$ ; (זה מותר ורצוי כשיודעים מראש איזה ערך רוצים)

## פלט

כיצד נציג את מה שנרצה על מסך הקונסול? על ידי פקודת:  
`Console.WriteLine( )`

מה שנרצה להציג יהיה בתוך הסוגריים. אם נרצה להציג מחרוזת, נקיף אותה בצמד של גרשיים. לדוגמה:

```
int x = 10;  
Console.WriteLine(x);
```

על המסך יופיע מספר 10.

עוד דוגמה:

```
string my_name = "Nemo";  
string animal_type = "Fish";  
Console.WriteLine("My name is: " + my_name + " and I am a " + animal_type);
```

מה יודפס? שימו לב שמה שמוקף בגרשיים מודפס בדיוק כמו שהוא, וסימן החיבור (+) למעשה מצמיד מחרוזות אחת לשניה, כך שמה שיתקבל זה:

My name is: Nemo and I am a Fish

אנו מחברים כאן שתי מחרוזות בין צמד גרשיים ושני משתנים שבתוכם נמצאות מחרוזות.

בודאי הבחנתם שעבור מספרים לא צריך להקיפם בצמד גרשיים. מה יודפס אם כן, כתוצאה מהקוד הבא:

```
int num = 10;  
double x = 20.4  
Console.WriteLine(num + " kilos are: " + x + " pounds");
```

חיברתי כאן בין מחרוזות ומספרים. המחשב יודע להסתדר עם זה (הוא מיסב באופן פנימי את המספרים למחרוזות ומצמיד אותם אחת לשניה). הוא עושה זאת רק עבור הפלט כאן, ולא ממש משנה את x למשל.

מה עושה הקוד הבא?

```
string x = "Wonder woman";  
string y = "Hello, I am ";  
string z = y+x;  
Console.WriteLine(z);
```

נסו לנחש, ואז בדקו על המחשב.