



# הוראות הדפסה ומשתנים

יסודות מדעי המחשב



## ביחידה זו נלמד:

- הדפסה למסך
- הגדרת משתנים
- קבלת מידע מהמשתמש
- מבנה הזיכרון
- השמת ערך למשתנים
- קבועים



## התוכנית הראשונה שלי

בשלב זה נתעלם מכל הטקסט שכתוב פה

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace first_project
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

כאשר נפתח את סביבת העבודה ליצירת פרויקט, נקבל קובץ שנראה כך:

Main היא פעולה שתכיל את אוסף ההוראות שלנו שיישמו בין ה {} - וזה נקודת ההתחלה של התוכנית שלנו

הריצה של כל תוכנית תפתח לנו חלון ותציג למסך חלק מהפקודות שבצענו.  
בתוכנית זו לא בצענו אף פקודה لكن תוכן החalon ריק.

Press any key to continue . . .



## תוכנית המדפסה למסך

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace first_project
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            Console.WriteLine("Hi!");
        }
    }
}
```

פקודת הדפסה למסך.  
תודפס למסך המחרוזת שבין הגרשיים

בסוף כל פקודה יש ;



## תוכנית המדפיס שורה למסך

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace first_project
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
            Console.WriteLine("Hi!");
        }
    }
}
```

ההצגה מראה קוד C# פשוט אשר מדפס שתי שורות של טקסט באמצעות הוראות `Console.WriteLine`. הקוד כולל imports עבור `System`, `System.Collections.Generic`, `System.Linq`, `System.Text` ו-`System.Threading.Tasks`. הוא מוגדר ב-`first_project` ו-`Program`. הוראות `Main` ו-`Console.WriteLine` מודפסות את הטקסטים "Hello World!" ו-"Hi!" בהתאמה.

ההצגה מוצגת כTerminal窗口, עם תוצאות המילוי:

```
Hello World!
Hi!
Press any key to continue . . .
```

פקודת הדפסת שורה למסך  
וירידת שורה בסופה

## אז מה זה תכנות ?

- קוד בשפת תכנות הוא קובץ טקסט הכתוב מאוסף מילים מסויימות באנגלית.
- יש מילים מסויימות שהן הוראות מוכראות למחשב.
- המחשב אינו יודע אנגלית, ולכן علينا לתרגם עבورو את הקוד שלנו, בתחילת הנקרא **קומpileציה** (הידור).

## קיבלה קלט מהמשתמש

- היינו רוצים שהתוכנית שלנו תהיה גמישה, ושתדע להדפיס את המחרוזת שראינו. עברו כל מחרוזת שתתקבל מהמשתמש שיריץ את התוכנית.
- לשם כך עליינו ללמידה הדברים הבאים:
  - ✓ כיצד מקבלים מידע מהמשתמש
  - ✓ היכן מאחסנים מידע זה
  - ✓ כיצד מדפיסים מידע זה



↶ ↻ ↺ ↻

8 of 41

< PREV

NEXT >

## קבלת קלט מהמשתמש : הפקודה והגדרת משתנה

- פקודה נוספת שהמחשב מכיר היא קבלת ערך מהמשתמש.

```
class Program
{
    static void Main(string[] args)
    {
        string x;
        x = Console.ReadLine();
    }
}
```

את הערך שנקלט נציב במשתנה x

פירוש הפקודה: `Console.ReadLine()`  
הוא: קלוט ערך מהמשתמש

- מה זה x ?
- x הוא שם המקום שבו נאחסן את הערך הנקלט מהמשתמש.
- כדי להקצות עבורנו מקום זה علينا להגדיר **משתנה**.
- מה זה **string** ?
- סוג של מקום בו ניתן לשמור מחרוזת (טקסט).



## אחסון ערכים

- כאשר מרכיבים תוכנית רצה לשמר משתנים מסוימים כדי שיישמשו אותן בפקודות בהמשך.
- כאשר התוכנית רצה, היא מקבלת שטח בזיכרון בו היא יכולה לאחסן משתנים אלו.

1000
1001
1002
1003
1004
1005

שטח זה נקרא מהשנית זמן ריצה (RunTimeStack).

- המשנית מורכבת מתאים, וכל תא יש כתובות כל תא נקרא בית (byte).
- יש משתנים שערכם מאוחסן בבית אחד, ויש משתנים שערכם מאוחסן ביותר (נראה בהמשך).



## אחסון ערכים : דוגמה

- בתוכנית זו הגדכנו את המשתנה x, ולכן התוכנית מבקשת תא במחסנית למשתנה זה.
- כאשר המשתמש מקליד מחרוזת, היא נכנסת לתוך התא המוקצה עבור x.

```
class Program
{
    static void Main(string[] args)
    {
        string x;

        x = Console.ReadLine(); ←
    }
}
```

string: x	"hey"	1000
		1004
		1008
		1012
		1016
		1020

משתנה מחרוזת צריך בית עבור כל תו.



&gt;



## הדף ערכו של משתנה

```
class Program
{
    static void Main(string[] args)
    {
        string x;

        Console.WriteLine("Enter a word: ");
        x = Console.ReadLine(); לאחר הקדחה ←
        Console.WriteLine("The word is " + x);
    }
}
```

```
Enter a word:  
hey  
The word is hey
```

string: x	"hey"	1000
		1004
		1008
		1012
		1016
		1020



C ↗ ⏪

## הגדרות

### הוראה

מילה שומרה שמשמעותה להוראות למחשב לבצע פעולה מסוימת.  
הוראות שאנו נזכיר כבר מכיריכם:

`Console.WriteLine("");`

✓ כתיבה למסך

`x = Console.ReadLine();`

✓ קבלת נתונים מהמשתמש

והצבתם במשתנה x

### פעולה

אוסף הוראות לביצוע ייחודית לוגית מסוימת.  
ה-`Main` הוא דוגמה לפעולה המכילה רצף הוראות לביצוע.  
בהמשך נראה עוד על כתיבה ושימוש בפעולות.



C

↶

↷

13 of 41

< PREV

NEXT >

```
class Program
{
    static void foo()
    {

    }

    static void Main(string[] args)
    {
        foo();           zman ha'avorah
    }
}
```

## פעולות

- רצף ההוראות ב פעולה הוא לפי סדר כתיבתן.
- רצף הפעולות בתוכנית הוא לפי סדר השימוש בהן (הזמןון שלהם) ב פעולה הראשית (Main).
- נעבד בסגנון שכל ההוראות שאנחנו כתבimos יהיו עטופות ב פעולה שאנחנו כתבimos.



&gt;



```
class Program
{
    static void foo()
    {

    }

    static void Main(string[] args)
    {
        foo();           זימון הפעולה

        string x;

        Console.WriteLine("Enter a number: ");
        x = Console.ReadLine();
        Console.WriteLine("The number is " + x);
    }
}
```

## פעולות

- רצף ההוראות ב פעולה הוא לפי סדר כתיבתן.
- רצף הפעולות בתוכנית הוא לפי סדר השימוש בהן (הזמן שלם) ב פעולה הראשית (Main).
- נעבד בסגנון שכל ההוראות שאנו חנו כתבים יהיו עטופות ב פעולה שאנו חנו כתבים.
- בעת זימון הפעולה, נקפוץ לאוסף הפעולות שלה ונבצע אותן.
- לאחר שנסיים עם הפקודות ב פעולה, נמשיך עם הפקודות הרשומות ב- Main.

## טיפוסי משתנים

- עד כה רأינו כיצד קולטים מהמשתמש נתונים באמצעות הוראה `( )`. `Console.ReadLine()`.
- הנתון שקראנו הוכנס לערך משתנה מטיפוס `string`, שיודע לאחסן מחרוזת (טקסט).
- ניתן להמיר את הנתון שקראנו למספרשלם, מספר עשרוני, בוליאני אותו.
- סוג הטיפוס הוא בהתאם לפעולות שנוכל לבצע על הטיפוס, למשל כפל יבצע על מספרים.
- יש לידע את המחשב מהו הטיפוס של הנתונים איתו אנחנו רוצים לעבוד.

```
static void foo()
{
    string x;

    Console.WriteLine("Enter a number: ");
    x = Console.ReadLine();
    Console.WriteLine("The number is " + x);
}
```



- ✓ משתנה הוא שטח אחסון עבור נתונים מסוימים.
- ✓ לכל משתנה גודל מסוים בהתאם לסוג המידע.



## שמירת המשתנים בזיכרון



## טיפוסי משתנים (Data Types)

ישנם סוגים שונים של משתנים.

המשתנים נבדלים ב:

1. **בסוג התוכן** אותו הם יכולים להכיל (שלם, עשרוני, טו, בוליани, מחרוזת).
2. **במספר** הבטים (bytes) שהמשתנה תופס בזכרון.

שם הטיפוס	סוג נתונים	גודל בזיכרון	דוגמאות
string	מחרוזת	הפניה, יוסבר בהמשך	"hi", "bla bla"
int	מספרשלם	4	-8, 2, 7
double	מספרעשרות	8	6.8, -8.2
char	טו	2	'a', 'N', '7', 'I'
bool	בוליани	1	true / false



&gt;



## הגדרת וקליטת נתונים מטיפוסים שונים

```
static void Main(string[] args)
{
    string name;
    int yearOfBirth, currentYear, age;
    double gradesAverage;
    char bestGrade;
    bool likeProgramming;

    Console.Write("Enter your name: ");
    name = Console.ReadLine();

    Console.Write("Enter your birth year: ");
    yearOfBirth = int.Parse(Console.ReadLine());

    Console.Write("What is the year now? ");
    currentYear = int.Parse(Console.ReadLine());

    Console.Write("Enter your grades average: ");
    gradesAverage = double.Parse(Console.ReadLine());

    Console.Write("Enter your best grade: ");
    bestGrade = char.Parse(Console.ReadLine());

    Console.Write("Do you like programming? ");
    likeProgramming = bool.Parse(Console.ReadLine());
}

age = currentYear - yearOfBirth;

Console.WriteLine("Hi " + name + "!");
Console.WriteLine("You are " + age + " years old");
Console.WriteLine("Your grades average is "
    + gradesAverage + " and your best grade is "
    + bestGrade);
Console.WriteLine("And do you like programming?? "
    + likeProgramming + "!!!");
```

```
Enter your name: gogo
Enter your birth year: 2004
What is the year now? 2021
Enter your grades average: 91.3
Enter your best grade: A
Do you like programming? True

Hi gogo!
You are 17 years old
Your grades average is 91.3 and your best grade is A
And do you like programming?? True!!!
```

## קליטת נתונים מטיפוסים שונים : סיכום הוראות

- קליטת נתונים היא תמיד קליטתו כ-`string` ומרתו לטיפוס המבוקש ע"י הפעולה `Parse`.
- הוראות הקלט, עבור הטיפוסים השונים:
  - `int.Parse(Console.ReadLine())` - `int`
  - `double.Parse(Console.ReadLine())` - `double`
  - `char.Parse(Console.ReadLine())` - `char`
  - `bool.Parse(Console.ReadLine())` - `bool`
- ההנחה שלנו, בשלב זה, תהיה שהקלט תקין. כמובן, אם ביקשנו מספר שלם, אך הוקלד מספר שלם, אחרת תתקבל שגיאה (נלמד לטפל בשגיאות בהמשך).



```
static void WrongInput()
{
    int num;

    Console.Write("Enter a number: ");
    num = int.Parse(Console.ReadLine());

    Console.WriteLine("The number is " + num);
}
```

## דוגמא לניסיונ המריה שיכשל

Enter a number: 7y

7y: קלט שגוי.  
המריה נכשלה

Enter a number: 82  
The number is 82

:82 קלט תקין

```
Unhandled Exception: System.FormatException: Input string was not in a correct format.
   at System.Number.StringToNumber(String str, NumberStyles options, NumberBuffer&
number, NumberFormatInfo info, Boolean parseDecimal)
   at System.Number.ParseInt32(String s, NumberStyles style, NumberFormatInfo info)
   at System.Int32.Parse(String s)
   at first_project.Program.WrongInput() in C:\Users\keran\OneDrive\private
business\מצגות להקלטה\משרד החינוך\חברות\Code\Output-Input and Variables\first
project\Program.cs:line 60
   at first_project.Program.Main(String[] args) in C:\Users\keran\OneDrive\private
business\מצגות להקלטה\משרד החינוך\חברות\Code\Output-Input and Variables\first
project\Program.cs:line 68
```



## הגדרת משתנים בתוכנית

שמות המשתנים יכולים להכיל אותיות גדולות וקטנות, מספרים וכן תחתון בלבד.

```
static void VariablesNames()
{
    int nos;           שם חסר משמעות
    int num_of_students;
    int numOfStudents;
    int student#1;
    int num_of_students;
    int 1student;
}

numOfStudents = 10;
numofstudents = 10;
```

- יש לתת שמותמשמעותיים !
- שם המשתנה לא יכול רוחים.
- לא ניתן להתחיל שם המשתנה במספר.
- יש הבדל בין אותיות גדולות לקטנות.

אין להשתמש במילים שמורות של השפה (נכיר בהמשך).

למען הקראות,  
כל מילה בשם המשתנה תתחיל באות גדולה.

## הגדרת משתנים בתוכנית : הזיכרון

```
static void ReadTypes()
{
    int yearOfBirth, currentYear, age;
    double gradesAverage;
    char bestGrade;
    bool likeProgramming;
}
```

int מרכיב מ-4 בתים

char מרכיב מ-2 בתים

int yearOfBirth		1000
int currentYear		1004
int age		1008
double gradesAverage		1012
char bestGrade		1020
bool likeProgramming		1022

- למה הפרש בין הכתובות שונה?
- כי כל סוג משתנה תופס מספר שונה של בתים

### הגדרת משתנים:

<type> <var name>[=init value][,<var name2>...];



## איך המידע שומר במחשב ?

כשהאנחנו סופרים אנחנו משתמשים ב-10 ספרות

- למה 10 ?
- נראה כי יש לנו 10 אצבעות...

באמצעות 10 ספרות ניתן לייצג את כל המספרים בעולם.

از הבחירה דואקן ב-10 ספרות הייתה שרירותית.

ובאופן דומה, הינו יכולים לייצג כל מספר בעולם גם למשל רק עם 8 ספרות.



למחשב אין 10 אצבעות, אבל הוא מכיר 2 מצבים: יש חשמל ואין חשמל.

זאת אומרת, שבאופן דומה, המחשב יכול לייצג כל מספר בעולם באמצעות שני ערכים בלבד.

הערכים שנבחרו לייצג את מצביו המחשב הם 0 ו-1, וזה נקראת **שפה בינארית**.

לא נרחיב פה על ההסביר, אבל רק נזכיר שלمثال עבור המחשב המספר 24 מיוצג כר 11000.



## אתחול והשנת משתנים

- כאשר מגדירים משתנה מהסוגים שתוארו במצגת, יש ליתת לו ערך התחלתי בזמן הגדרכתו או במשפט השמה.
- אם מנסים להשתמש בערך של משתנה שלא אוחחל - מתקבלת שגיאת קומפיילציה !

```
static void InitAndAssignVariables()
{
    int x, y, z = 9;
    x = 5;
    y = x * 3;
}
```

פעולות השמה היא  
נתן ערך למשתנה  
ראשית מחושב הביטוי הצד ימין  
ו吐צאתו מושמת במשתנה שימושה

int x	5	1000
int y	15	1004
Int z	9	1008

## אתחול והשנת משתנים מטיפוס מחרוזת

- כאשר מגדירים משתנה מטיפוס מחרוזת, יש לאותחלו לפני השימוש, אחרת ערכו לא מוגדר.
- עבודה עם מחרוזת שאינה מאותחלת גורר שגיאות קומpileציה.

```
static void InitAndAssignStrings()
{
    string str;
    Console.WriteLine(" | " + str + " | ");
    str = "hello";
    Console.WriteLine(" | " + str + " | ");
}
```



|hello|

## השمات בין טיפוסים שונים

```
static void ConversionRules()
{
    int n1 = 3;
    double d = 97.2;
    string s = "45.6";
    n1 = (int)d;
    d = (double)n1;
    d = double.Parse(s);
    s = n1 + "";
}
```

בашמה של מספר מטיפוס `double` לтип `int`  
מאנדרים את השבר  
ומקבלים רק את החלם.  
השמה זו דורשת המרה מפורשת (`int`).

בашמה של מספר מטיפוס `int` לтип `double`  
אין איבוד מידע  
ולכן אין חובה להשתמש בהמרה (`double`).

שרשור של מספר למחרוזת  
כולל בתוכו המרה `string`-ל-

המרה של מחוץ, למשל לтип `double`,  
תתבצע ע"י שימוש בפעולה `Parse`.

<code>int</code> n1	97	1000
<code>double</code> d	45.6	1004
<code>string</code> s	"97"	1012



&gt;



Decimal	Char	Decimal	Char	Decimal	Char
32	(space)	64	Ø	96	ׁ
33	!	65	܂	97	܃
34	"	66	܄	98	܅
35	#	67	܆	99	܇
36	\$	68	܈	100	܉
37	%	69	܊	101	܋
38	&	70	܌	102	܍
39	'	71	܏	103	ܑ
40	(	72	ܒ	104	ܓ
41	)	73	ܔ	105	ܕ
42	*	74	ܖ	106	ܗ
43	+	75	ܘ	107	ܙ
44	,	76	ܚ	108	ܚ
45	-	77	ܛ	109	ܜ
46	.	78	ܝ	110	ܞ
47	/	79	ܟ	111	ܠ
48	ܠ	80	ܡ	112	ܢ
49	ܡ	81	ܣ	113	ܤ
50	ܢ	82	ܥ	114	ܦ
51	ܦ	83	ܨ	115	ܪ
52	ܫ	84	ܩ	116	ܪ
53	ܪ	85	ܪ	117	ܪ
54	ܪ	86	ܪ	118	ܪ
55	ܪ	87	ܪ	119	ܪ
56	ܪ	88	ܪ	120	ܪ
57	ܪ	89	ܪ	121	ܪ
58	:	90	ܪ	122	ܪ
59	:	91	[	123	{
60	<	92	\	124	
61	-	93	]	125	}
62	>	94	^	126	"
63	?	95		127	

ערך ה-ASCII של האותיות  
הגדולות נמצאים ברכף

ערך ה-ASCII של האותיות  
הקטנות נמצאים ברכף

## יצוג תווים

- למדנו שהמחשב מכיר רק שני סימנים, באמצעותם ניתן לייצג כל מספר.
- המחשב אינו מכיר תווים, لكن לכל تو יש קוד מספרי הנקרא קוד ASCII.
- ניתן לראות את הקוד של כל تو בטבלה ASCII.
- כארט מאחסנים את ערכו של משתנה מטיפוס char למעשה שומרים את ערך ה-ASCII שלו.

### ASCII:

American Standard for Code Information Interchange

ערך ה-ASCII של הספרות נמצאים ברכף.  
שימו לב כי ערך ה-ASCII של ספרה אינה זהה לערכה המספרית !

## הגדרת משתנים מטיפוס char

- משתנה מטיפוס `char` מכיל תו, ולכן ניתן לשים בו תו, ואפשר גם מספר.
- כאשר שמים במשתנה מספר הכוונה לטו שמספר זה מייצג בטבלת ה-ASCII.
- כדי לקבל את התו, יש לבצע המרה יוזמה לטיפוס `char`.
- בשני המקרים הללו הערך בתא הזיכרון זהה (מספר שערך ה-ASCII של 'a' הוא 97).

```
static void RepresentChar()
{
    char ch1 = 'a';
    char ch2 = (char)97;

    Console.WriteLine("ch1=" + ch1 + ", ch2=" + ch2);
}
```

ch1=a, ch2=a

char ch1	01100001	1000
char ch2	01100001	1002
		1004

Decimal	Char
32	(space)
33	!
34	"
35	#
36	\$
37	%
38	&
39	'
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	:
60	<
61	=
62	>
63	?

```
static void RepresentChar()
{
    char ch1 = '8';
    char ch2 = (char)8;

    Console.WriteLine("ch1=" + ch1 + ", ch2=" + ch2);
}
```

ch1=8, ch2=

char ch1	<b>01010110</b>	1000
char ch2	<b>00001000</b>	1001
		1002

## הגדרת משתנים מטיפוס char

- במקרה של ch1 ערכו '8', כלומרתו המיצג את הספרה 8 (בגלל שהוא בתוך ').
- ערך ה-ASCII של התו '8' (56) נשמר בשפת המחשב, ולא 8 בבינארית.
- לעומתו ערכו של ch2 הוא המספר 8 (זההתו שהוא לא "דפס").

```
static void CreateNumberFromDigits()
{
    int digit1, digit2, digit3;
    int newNum = 0;

    Console.Write("Enter the hundreds digit: ");
    digit1 = int.Parse(Console.ReadLine());

    Console.Write("Enter the dozens digit: ");
    digit2 = int.Parse(Console.ReadLine());

    Console.Write("Enter the unit digit: ");
    digit3 = int.Parse(Console.ReadLine());

    newNum = digit1 * 100 + digit2 * 10 + digit3;

    Console.WriteLine("The number is " + newNum);
}
```



## תרגיל: קליטת ספרות

- קלטו 3 ספרות:
  - הראשונה תהיה ספרת המאות
  - השנייה ספרת העשרות
  - השלישית ספרת האחדות
- צרו משתנה מטיפוס int שיחזיק את המספר התלת ספרתי שהוזן.

```
Enter the hundreds digit: 3
Enter the dozens digit: 4
Enter the unit digit: 8
The number is 348
```

Decimal	Char
32	(space)
33	!
34	"
35	#
36	\$
37	%
38	&
39	,
40	(
41	)
42	*
43	+
44	:
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	:
60	<
61	=
62	>
63	?

```
static void CreateNumberFromChars()
{
    char digit1, digit2, digit3;
    int newNum = 0;

    Console.Write("Enter the hundreds digit: ");
    digit1 = char.Parse(Console.ReadLine());

    Console.Write("Enter the dozens digit: ");
    digit2 = char.Parse(Console.ReadLine());

    Console.Write("Enter the unit digit: ");
    digit3 = char.Parse(Console.ReadLine());

    51*100 + 52*10 + 56 = 5676
    '3'*100 + '4'*10 + '8'
    newNum = digit1 * 100 + digit2 * 10 + digit3;

    Console.WriteLine("The number is " + newNum);
}
```

## תרגיל: קליטת תווים כמספרות

קלטנו 3 תווים :

- הראשון יהיה ספרת המאות
- השני ספרת העשרות
- השלישי ספרת האחדות

צרו משתנה מטיפוס int שיכזיק את המספר התלת ספרתי שהוזן.

```
Enter the hundreds digit: 3
Enter the dozens digit: 4
Enter the unit digit: 8
The number is 5676
```

Decimal	Char
32	(space)
33	!
34	"
35	#
36	\$
37	%
38	&
39	'
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	:
60	<
61	=
62	>
63	?

```
static void DigitFromLetter()
{
    char ch = '8';

    Console.WriteLine(ch+3);
    Console.WriteLine((char)(ch + 3));
    int digit = ch - '0';
    Console.WriteLine(digit);
    Console.WriteLine(digit+3);
}
```

59  
;  
8  
11

- עברו התו '8', ערך ה-ASCII שלו הוא 56.
- נחסר ממנו את ערך ה-ASCII של התו '0', שהוא 48.
- התוצאה:  $56-48=8$ .

## טריק: קבלת ערך של ספרה מຕוךתו

- בהינתן משתנה מטיפוס `char` שמאוחסנת בו ספרה, למשה מאוחסן בו ערך (בין 48 ל-57) מתוך ערכי ה-ASCII של הספרות.
- כדי לקבל את ערך הספרה, ולא את ערך התו ASCII שלה, נחסר מהמשתנה את ערך ה-ASCII של הספרה '0'.

דוגמה:

## תרגיל: קליטת תוים מספרות - תיקון

Decimal	Char
32	(space)
33	!
34	"
35	#
36	\$
37	%
38	&
39	,
40	(
41	)
42	*
43	+
44	:
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	:
60	<
61	=
62	>
63	?

```
static void CreateNumberFromChars()
{
    char digit1, digit2, digit3;
    int newNum = 0;

    Console.Write("Enter the hundreds digit: ");
    digit1 = char.Parse(Console.ReadLine());

    Console.Write("Enter the dozens digit: ");
    digit2 = char.Parse(Console.ReadLine());

    Console.Write("Enter the unit digit: ");
    digit3 = char.Parse(Console.ReadLine());

    newNum = (digit1-'0') * 100 + (digit2-'0') * 10 + (digit3-'0');

    Console.WriteLine("The number is " + newNum);
}
```

```
Enter the hundreds digit: 3
Enter the dozens digit: 4
Enter the unit digit: 8
The number is 348
```

## הדף תווים מיוחדים

```
static void PrintSpecialChars()
{
    Console.Write("Hello World!\nNew Line :-)");
    Console.WriteLine("Hello\tWorld!");
    Console.WriteLine("\"Hello World!\"");
    Console.WriteLine("\\Hello World!");
}
```

```
Hello World!
New Line :-)Hello      World!
"Hello World!"
\Hello World!
```

ירידת שורה	\
\t	tab
\"	"
\\	\

## דוגמה: מחיר+מע"מ של מוצר

```
static void ProductPrice()
{
    double price;
    double totalPrice;

    Console.WriteLine("Please enter the product's price: ");
    price = double.Parse(Console.ReadLine());

    totalPrice = price * 1.17; // price + (price * 0.17);

    Console.WriteLine("The price including 17% tax is " + totalPrice);
}
```

- עבור כל מוצר שקונים צריך לשלם מע"מ, שערכו נקבע ע"י המדינה ומשתנה כל כמה שנים.
- נכתב תוכנית הקולטת מחיר מוצר ומציגה את מחירו לאחר תוספת המע"מ.

```
Please enter the product's price: 10
The price including 17% tax is 11.7
```

## מחיר+מע"מ של מוצר: הגדרת קבועים

```
static void ProductPrice()
{
    double price;
    double totalPrice;
    double TAX = 17; // שימוש במשתנה התוכניתית יותר קרייה,
                      // ומשמעות המספר ברורה
                      // גם למי שלא ידוע מהו ערךו של המע"מ

    Console.WriteLine("Please enter the product's price: ");
    price = double.Parse(Console.ReadLine());

    totalPrice = price + (price * TAX/100);

    //Console.WriteLine("The price including 17% tax is " + totalPrice);
    Console.WriteLine("The price including " + TAX + "% tax is " + totalPrice);
}
```

Please enter the product's price: 10  
The price including 17% tax is 11.7

שימוש במשתנה התוכניתית יותר קרייה,  
וממשמעות המספר ברורה  
גם למי שלא ידוע מהו ערכו של המע"מ

שינוי בערך המע"מ יגרור שינוי  
רק במקום אחד בלבד בתוכנית,  
מה שcosaר עבודה וביעות בעתיד  
(אם שכחנו לתקן במקום אחד...)



## קבועים

נהוג לנכטו שמות  
קבועים באותיות גדולות

לא ניתן לשנות ערך  
קבוע, בזמן ריצה

```
static void ProductPrice()
{
    double price;
    double totalPrice;
    const double TAX = 17;

    Console.WriteLine("Please enter the product's price: ");
    price = double.Parse(Console.ReadLine());

    TAX = 18.5;
    totalPrice = price + (price * TAX/100);

    Console.WriteLine("The price including " + TAX + "% tax is " + totalPrice);
}
```

בתוכנית זו ערכו של המע"מ **קבוע**  
ואינו נקבע במהלך ריצת התוכנית

יתרונות לשימוש בקבועים:  
1 - תכנית קרייה יותר  
2 - תחזוקה

- ערך קבוע נקבע בזמן קידוד הקוד ע"י המተכנת.
  - כדי לשנות את ערכו של הקבוע צריך לעדכן את הקוד ולקמפל את הקוד מחדש.
  - ברמת העקרונית נוכל לשנות את ערך המע"מ בפקודה אחרת.
  - כדי למנוע זאת נגדיר אותו כקבוע **const**.
- אם נרצה לאפשר שינוי ערך המע"מ בזמן ריצת התוכנית, נגדירו כ משתנה רגיל.



## דוגמה: מחיר נסיעה במנונית

- מחיר נסיעה במנונית משוקל מרכיבים הבאים:
  1. מחיר התחלתי: 10.20 ש"ח
  2. מחיר עבור כל ק"מ: 1.30 ש"ח
  3. מחיר עבור כל מזודה: 2.00 ש"ח
- יש לבקש מהמשתמש את הנתונים הדרושים ולהציג את הסכום לתשלום.

### מיפוי הנתונים:

1. מחיר התחלתי - נתון קבוע
2. מחיר עבור ק"מ - נתון קבוע
3. מחיר עבור מזודה - נתון קבוע
4. מספר ק"מ לנסעה - נתון קלט
5. מספר מזודות - נתון קלט
6. סה"כ לתשלום - נתון חישוב ונתון פلت



## מחיר נסיעה במנועית: אסטרטגיית הפתרון

### הנתונים הדרושים:

מחיר התחלתי – נתון קבוע  
 מחיר עبور ק"מ – נתון קבוע  
 מחיר עبور מזודה – נתון קבוע  
 מספר ק"מ לנסעה – נתון קלט  
 מספר מזודות – נתון קלט  
 סה"כ לתשלום – נתון חישוב ונחתן פלט

מחיר נסעה במנועית משוקלל מהרכיבים הבאים:

מחיר התחלתי: 10.20 ש"ח

מחיר עبور כל ק"מ: 1.30 ש"ח

מחיר עبور כל מזודה: 2.00 ש"ח

- גדר START\_FEE = 10.20
- גדר PRICE\_PER\_KM = 1.30
- גדר PRICE\_PER\_LUGGAGE = 2.00
- נקלוט מהמשתמש את מספר הק"מ לטור numOfKm
- נקלוט מהמשתמש את מספר המזודות לטור numOfLuggages
- Calculates the total sum for payment:  
total = START\_FEE + numOfKm \* PRICE\_PER\_KM + numOfLuggages \* PRICE\_PER\_LUGGAGE
- Shows the total value:  
Nazig at total.

שים לב שהנוסחה עבדת גם אם מספר המזודות הוא 0



```
static void TaxiDrivePrice()
{
    const double START_FEE = 10.20;
    const double PRICE_PER_KM = 1.30;
    const double PRICE_PER_LUGGAGE = 2.00;

    double numOfKm;
    int numOfLuggages;
    double total;

    Console.Write("How many kilometers? ");
    numOfKm = double.Parse(Console.ReadLine());

    Console.Write("How many luggages? ");
    numOfLuggages = int.Parse(Console.ReadLine());

    total = START_FEE + numOfKm * PRICE_PER_KM + numOfLuggages * PRICE_PER_LUGGAGE;

    Console.WriteLine("You need to pay " + total + "NIS");
}
```

## מחיר נסיעה במונייה: הקוד

1. נגידר `START_FEE = 10.20`
2. נגידר `PRICE_PER_KM = 1.30`
3. נגידר `PRICE_PER_LUGGAGE = 2.00`
4. נקלוט מהמשתמש את מספר הק"מ למשך `numOfKm`
5. נקלוט מהמשתמש את מספר המזוודות לתוך `numOfLuggages`
- 6.-Calculates the total price:  
$$\text{total} = \text{START\_FEE} + \text{numOfKm} * \text{PRICE\_PER\_KM} + \text{numOfLuggages} * \text{PRICE\_PER\_LUGGAGE}$$
7. Prints the `total`



## ביחידה זו למדנו:

- הדפסה למסך
- הגדרת משתנים
- קבלת מידע מהמשתמש
- מבנה הזיכרון
- שימוש ערך למשתנים
- קבועים

