

תקציר הוראות השפה - C#

טיפוסים משתנים בסיסיים (פרימיטיבים) בשפת C#:

int	(בתחום $\pm 2,147,483,648$)	מספרי שלם
long		מספרי שלם ארוך
double		מספר ממשי
char		תו
string		מחרוזת

<p>const ערך-הקבוע = שם-הקבוע טיפוס-נתונים</p> <p>const int N = 10; const double X = 2.34;</p> <p>int.MaxValue // הערך השלם הגדול ביותר int.MinValue // הערך השלם הקטן ביותר</p>	<p>קבועים:</p> <p>דוגמא: נהוג לרשום שמות קבועים באות גדולה</p> <p>קבועים בשפה:</p>
---	--

<p>int a, b; double x, y;</p>	הגדרת משתנים:
---	----------------------

מבנה התכנית בשפת C#:

<pre>public class שם-המחלקה { public static void Main(string[] args) { הגדרת המשתנים; הוראה; הוראה; } }</pre>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> שים לב להזחת identification (קפיצת tab של תחילת השורה) </div>
--	--

הערות

// הערת שורה	
/* הערה */	הערת קטע הנמשכת על פני כמה שורות

• ב-C# יש חשיבות לגודל אות (case sensitive)

קלט/פלט:

<pre>Console.Write ("מחרוזת בשפה האנגלית"); Console.WriteLine ("מחרוזת");</pre>	פלט מחרוזת מחרוזות יוצגו בשפה האנגלית
<pre>Console.Write (שם-משתנה); Console.Write (שם-משתנה + " " + שם-משתנה); Console.WriteLine (שם-משתנה); Console.WriteLine (שם-משתנה + " " + שם-משתנה); Console.WriteLine ("שם משתנה" + "שם משתנה");</pre>	הדפסת תוכן משתנה פורמט \$ החל מגרסת 2017
<pre>Console.Write (שם-משתנה + " מחרוזת "); Console.Write (" מחרוזת " + שם-משתנה + " מחרוזת"); Console.WriteLine (שם-משתנה + " מחרוזת "); Console.WriteLine ("מחרוזת" + שם-משתנה + "מחרוזת"); Console.WriteLine ("מחרוזת {שם-משתנה} מחרוזת");</pre>	הדפסת משתנים ומחרוזות
<pre>Console.WriteLine(\$"x = {x}, num = {num}"); // {x} x תוכן משתנה</pre>	הדפסת משתנים ומחרוזות
<pre>int a = int.Parse(Console.ReadLine());</pre>	קלט מספר שלם
<pre>double x = double.Parse(Console.ReadLine());</pre>	קלט מספר ממשי
<pre>char tav = char.Parse(Console.ReadLine()); char ch = Convert.ToChar(Console.ReadLine());</pre>	קלט מספר תו או :
<pre>string str = Console.ReadLine ();</pre>	קלט מחרוזת

<ul style="list-style-type: none"> בכל הוראת קלט ניתן לקלוט בדיוק משתנה אחד. לפני כל הוראת קלט יש להציג בקשה: <pre>Console.Write ("type a number → "); int a = int.Parse(Console.ReadLine());</pre>	דגשים
---	-------

הוראות השמה:

תו	מחרוזת	מספר	
<code>char ch = ' '; // תו רווח</code>	<code>string str = " "; // מחרוזת ריקה</code>	<code>int a = 0;</code>	אתחול בהגדרה
<code>ch = 'b';</code>	<code>str = "hello";</code>	<code>a = 5;</code>	הצבת קבוע

פעולות חישוביות:

<code>int a, b, c;</code> <code>double x, y, z;</code>	נתונים המשתנים הבאים:
<code>c = a + b;</code>	חיבור
<code>c = a - b;</code>	חיסור
<code>c = a * b;</code>	כפל
שלם / שלם ← שלם ממשי / ממשי ← ממשי שלם / ממשי ← ממשי שלם / שלם (double) ← ממשי	חילוק:
<code>z = x / y;</code> // לפחות אחד המשתנים המשתתפים // בפעולה החישוב חייב להיות ממשי <code>z = (double)a / b;</code>	חילוק בממשיים • המרה מפורשת
<code>c = a / b;</code>	חילוק בשלמים (מנה)
<code>c = a % b;</code>	שארית
• <u>המרה</u> - ניתן להמיר משלם לממשי ולהיפך בהמרה מפורשת (casting).	

קיצורים:

<code>a ++;</code>	הגדלה עצמית
<code>a --;</code>	הקטנה עצמית

הוראות השפה:

שונה <code>!=</code>	שווה <code>==</code>	סוגי יחס (קרא משמאל לימין)
גדול או שווה <code>>=</code>	גדול מ- <code>></code>	
קטן או שווה <code>>=</code>	קטן מ- <code><</code>	

בלוק הוראות:

	{ בלוק הוראות }	
if (a > 5) b = b + 1;	if (פסוק-לוגי) ביצוע;	אם ... (הוראה יחידה)
if (a == 3) { c = b * 2; Console.WriteLine ("result : " + c); }	if (פסוק-לוגי) { הוראה; הוראה; }	אם ... (בלוק הוראות)
if (a != 5) b = b + 1; else b = b - 1;	if (פסוק-לוגי) ביצוע; else ביצוע;	אם ... אחרת ... (הוראה יחידה)
if (a > 3) { Console.Write ("a number →"); b = int .Parse(Console.ReadLine()); } else { b = a; c = c + 1; }	if (פסוק-לוגי) { הוראה1; הוראה2; } else { הוראה3; הוראה4; }	אם ... אחרת ... (בלוק הוראות)
switch (a) { case 1: Console.WriteLine("one"); break; case 2: Console.WriteLine("two"); break; case 3: case 4: case 5: Console.WriteLine("five"); break; case 6: Console.WriteLine("six"); break; default : Console.WriteLine("out of range"); break; }	switch (משתנה) { case 1-ערוך: ביצוע-1; break; case 2-ערוך: ביצוע-2; break; case 3-ערוך: הוראה; הוראה; break; case 4-ערוך: ביצוע-4; break; default : ביצוע-5; break; }	ברירת החלטה • המשתנה הנבדק הוא מטיפוס סדור: מספר שלם או תו ההוראות מתבצעות עד ה- break ניתן לבחון רק ערכים בדידים
הפלט עבור 3, 4, ו-5 יהיה זהה five		

קשרים לוגיים

&&	וגם
	או
!	לא
<code>if (a > 3 && (b == 5 c != 0)) ...</code>	

משתנה בוליאני:

השמה למשתנה בוליאני:	
<pre>bool b = (num1 > num2); bool b; if (num1 > num2) b = true ; else b = false ;</pre>	<p>ביטויים שקולים:</p>
<code>bool found = ... ; // true או false ;</code>	
<pre>if (found) ... if (found == true) ...</pre>	<pre>if (! found) ... if (found == false) ...</pre>

לולאות:

<pre> for (קידום-אינדקס ; ערך-סיום <= אינדקס ; ערך-תחילי = אינדקס) { הוראה; הוראה; } ----- int a, sum = 0; for (i = 1 ; i <= 5 ; i++) { Console.Write ("a number → "); a = int.Parse(Console.ReadLine()); sum = sum + a; } </pre>	<p>לולאת for (לולאה עולה)</p>
<pre> for (הקטנת-אינדקס ; ערך-סיום >= אינדקס ; ערך-תחילי = אינדקס) { הוראה; הוראה; } ----- double x, sum = 0; for (i = 5 ; i >= 1 ; i --) { Console.Write ("a real number → "); x = int.Parse(Console.ReadLine()); sum = sum + x; } Console.WriteLine ("sum is: " + sum); </pre>	<p>לולאת for (לולאה יורדת)</p>
<pre> while (ביטוי-לוגי) { הוראה; הוראה; } ----- int sum = 0; Console.Write (" מספר ראשון → "); int a = int.Parse(Console.ReadLine()); while (a != 999) { sum = sum + a; Console.Write (" עוד מספר → "); a = int.Parse(Console.ReadLine()); } Console.WriteLine (sum + " : סכום המספרים"); </pre>	<p>לולאת while</p> <p>במחברת הבחינה מותר לכתוב מחרוזות בעברית, אלא אם כן נאמר אחרת</p>

פונקציות מתמטיות:

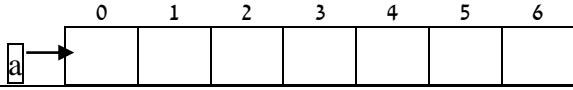
דוגמא	תחביר	מתקבל	מוחזר	טיפוס הערך
<code>int b = Math.Abs(a);</code> <code>double y = Math.Abs(x);</code>	<code>Math.Abs(x)</code>	שלם ממשי	שלם ממשי	ערך מוחלט $ x $
<code>if (Math.Sqrt(x) > y) ...</code>	<code>Math.Sqrt(x)</code>	שלם ממשי	ממשי	שורש ריבועי \sqrt{x}
<code>double p = Math.Pow(x, b);</code>	<code>Math.Pow(x, y)</code>	שלם ממשי	ממשי	חזקה x^y
<code>double x = Math.Round(7.853);</code> <code>int n = (int)Math.Round(3.456);</code>	<code>Math.Round(x)</code>	ממשי	ממשי	עיגול לשלם הקרוב בייצוג ממשי $4.0 \leftarrow 3.67$
<code>double x = Math.Round(7.856, 1);</code>	<code>Math.Round(x, d)</code>	ממשי	ממשי	עיגול בדיוק עשרוני של d ספרות: ($7.9 \leftarrow 7.856, 1$) ($7.86 \leftarrow 7.856, 2$)
<code>if (x == (int)x) ...;</code>	<code>(int)x</code>	ממשי	שלם	החלק השלם ($3 \leftarrow 3.67$)
<code>int small = Math.Min(12, a);</code>	<code>Math.Min(x, y)</code>	שלם ממשי	שלם ממשי	הערך הקטן מבין השניים
<code>int big = Math.Max(12, a);</code>	<code>Math.Max(x, y)</code>	שלם ממשי	שלם ממשי	הערך הגדול מבין השניים

מספר אקראי - שימוש באובייקט Random

<code>Random rnd = new Random();</code>	יצירת עצם:	בתוך גוף בתכנית:
<code>x = rnd.Next(n);</code>		מספר שלם בתחום 0 עד n (לא כולל) $0 \leq x < n$
<code>x = rnd.Next(from, to);</code>		מספר שלם בתחום from עד to (לא כולל) $from \leq x < to$

פעולות

<pre>static void ProcName (פרמטרים) { } </pre>	<p>נהוג ששם פעולה מתחיל באות גדולה פעולה שאינה מחזירה ערך</p>
<pre>static void Aaa (int a, int b, double x) { int i; : } </pre>	<p>(מוחזר ערך void)</p>
<pre>static FuncName (פרמטרים) טיפוס- מוחזר { : return ערך להחזרה; } </pre>	<p>פעולה המחזירה ערך</p>
<pre>static double Sum (int a, int b, double x) { double total ; total = a + b + x; return total; } </pre>	

<code>arr = new [10] טיפוס-נתונים</code>	מערך חד-ממדי
<code>int [] a = new int [7];</code> <code>double [] x = new double [N];</code> <code>char [] arrChar = new char [25];</code>	הגדרה
<code>x[2] = 2 * a[0];</code>	פנייה לתא
	
<pre>for (int i = 0 ; i < a.Length ; i++) a[i] = i;</pre> <p>גודל המערך הוא <code>a.Length</code> שם-המערך מספר התא האחרון במערך הוא <code>a.Length - 1</code></p>	סריקת מערך

<code>arr = new [,] טיפוס-נתונים [5,7];</code>	מעריך דו-ממדי (מטריצה)
<code>int [,] mat = new int [6,7];</code> <code>double [,] x = new double [N,M];</code>	
<p>סכום התא ה"ראשון" והתא ה"אחרון" במערך בגודל 6 שורות ו-7 עמודות:</p> <code>mat[3,4] = mat [0,0] + mat [5,6];</code>	פניה לתא
<p><code>mat.GetLength (0)</code> ⇒ מספר השורות במטריצה <code>mat.GetLength (1)</code> ⇒ מספר העמודות במטריצה</p> <pre>for (int i = 0 ; i < mat.GetLength (0) ; i++) for (int j = 0 ; j < mat.GetLength (1) ; j++) mat [i,j] =</pre>	סריקת המערך

string str ;	מחרוזות
str = "bla-bla-bla" ; str = new string ("bla-bla-bla") ;	מחרוזות ב-C# היא אובייקט
string str = Console.ReadLine();	קלט
פעולות שאינן משנות את המחרוזות	
num = str.Length;	אורך המחרוזות
str = str1 + str2 + "aaa" ;	שרשור
str1 == str2 // מחרוזות שוות ? str1 != str2 // מחרוזות שונות ? bool equal = str1.Equals(str2) ; int n = str1.CompareTo (str2) ; str1 > str2 ⇔ n > 0 str1 < str2 ⇔ n < 0 str1 == str2 ⇔ n = 0	השוואה
ch = str [i] ; 0 ≤ i < str.Length	תו i במחרוזת
int place = str.IndexOf (ch) ; // חיפוש הראשון	חיפוש תו
int place = str.IndexOf (subStr); // חיפוש הראשון	חיפוש תת-מחרוזת
הערה: indexOf מחזיר תמיד את הראשון מתחילת המחרוזת. אם לא נמצא יוחזר -1	
int place = str.IndexOf (ch, fromPlace); // חיפוש ממקום int place = str.IndexOf (subStr, fromPlace);	חיפוש המופע הבא:
int place = str.LastIndexOf (ch); // חיפוש מהסוף int place = str.LastIndexOf (subStr);	חיפוש מהסוף:
int place = str.LastIndexOf (ch, from); // חיפוש מהסוף ממקום int place = str.LastIndexOf (subStr, from);	