

תקציר הוראות השפה - C#

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

<p>int a, b;</p> <p>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>	<p>שים לב להזחת identification (קפיצת tab של תחילת השורה)</p>
--	---

הערות

// הערת שורה	
/* הערה */	<p>הערת קטע הנמשכת על פני כמה שורות</p>

• 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);</pre>	הדפסת תוכן משתנה
<pre>Console.Write (שם-משתנה + " מחרוזת "); Console.Write (" מחרוזת " + שם-משתנה + " מחרוזת"); Console.WriteLine (שם-משתנה + " מחרוזת "); Console.WriteLine ("מחרוזת" + שם-משתנה + " מחרוזת");</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)Console.Read(); char ch = Convert.ToChar(Console.Read());</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 = ' '; // 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:

<pre>int a, b, c; double x, y, z;</pre>	נתונים המשתנים הבאים :
<code>c = a + b;</code>	חיבור
<code>c = a - b;</code>	חיסור
<code>c = a * b;</code>	כפל

חילוק : שלם / שלם ← שלם ממשי / ממשי ← ממשי שלם / ממשי ← ממשי שלם / שלם (double) ← ממשי	
$z = x / y;$ // לפחות אחד המשתנים המשתתפים בפעולה החישוב חייב להיות ממשי // $z = (\text{double})a / b;$	חילוק בממשיים • המרה מפורשת
$c = a / b;$	חילוק בשלמים (מנה)
$c = a \% b;$	שארית
• <u>המרה</u> - ניתן להמיר משלם לממשי ולהיפך בהמרה מפורשת (casting).	

קיצורים :

$a ++;$	הגדלה עצמית
$a --;$	הקטנה עצמית

הוראות השפה :

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

בלוק הוראות :

	{ בלוק הוראות }
--	-----------------------

$\text{if } (a > 5)$ $\quad b = b + 1;$	(פסוק-לוגי) if ביצוע;	אם ... (הוראה יחידה)
$\text{if } (a == 3)$ $\{$ $\quad c = b * 2;$ $\quad \text{Console.WriteLine ("result : " + c);}$	(פסוק-לוגי) if $\{$ הוראה; הוראה; $\}$	אם ... (בלוק הוראות)
$\text{if } (a != 5)$ $\quad b = b + 1;$ else $\quad b = b - 1;$	(פסוק-לוגי) if ביצוע1; else ביצוע2;	אם ... אחרת ... (הוראה יחידה)

<pre> if (a > 3) { Console.Write ("a number → "); b = int.Parse(Console.ReadLine()); } else { b = a; c = c + 1; } </pre>	<pre> if (פסוק-לוגי) { 1הוראה; 2הוראה; } else { 3הוראה; 4הוראה; } </pre>	<p>אם... אחרת... (בלוק הוראות)</p>
<pre> 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; } </pre> <p>הפלט עבור 3, 4, ו-5 יהיה זהה five</p>	<pre> switch (משתנה) { case 1-ערך : ביצוע-1; break; case 2-ערך : ביצוע-2; break; case 3-ערך : הוראה; הוראה; break; case 4-ערך : ביצוע-4; break; default : ביצוע-5; break; } </pre>	<p>ברירת החלטה</p> <ul style="list-style-type: none"> המשתנה הנבדק הוא מטיפוס סדור: מספר שלם או תו ההוראות מתבצעות עד ה-break ניתן לבחון רק ערכים בדידים

קשרים לוגיים

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

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>
bool found = ... ; // true או false ;	
<pre>if (found) ... if (found == true) ...</pre>	<p>ביטויים שקולים</p>
<pre>if (! found) ... if (found == false) ...</pre>	<p>ביטויים שקולים</p>

לולאות:

<pre>for (קידום-אינדקס ; ערך-סיום <= אינדקס ; ערך-תחילי = אינדקס) { הוראה; הוראה; } ----- int a, sum = 0; for (int 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 (int 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:

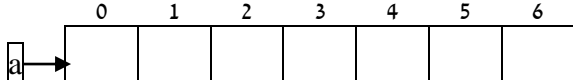
דוגמא	תחביר	מתקבל	מוחזר	טיפוס הערך
<code>int b = Math.Abs(a); double y = Math.Abs(x);</code>	<code>Math.Abs(x)</code>	שלם ממשי	שלם ממשי	ערך מוחלט $ x $
<code>if (Math.Sqrt (x) > y) ...</code>	<code>Math.Sqrt (x)</code>	שלם ממשי	ממשי	שורש ריבועי \sqrt{x}
<code>double p = Math.Pow (x, b);</code>	<code>Math.Pow (x, y)</code>	שלם ממשי	ממשי	חזקה x^y
<code>Console.WriteLine ((int)Math.Round(7.853));</code>	<code>(int) Math.Round (x)</code>	ממשי	שלם ארוך	עיגול לשלם הקרוב ($4 \leftarrow 3.67$)
<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 (שיטות method)

<pre>static void MethodName (פרמטרים) { } </pre>	<p>נהוג ששם פעולה מתחיל באות גדולה</p> <p>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 MethodName (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>int [] arr = {2, 5, 7, -1, 24};</code>	אתחול בהגדרה
<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>	מערך 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> <pre>mat[3,4] = mat [0,0] + mat [5,6];</pre>	פניה לתא
<pre>mat.GetLength (0) ⇨ מספר השורות במטריצה mat.GetLength (1) ⇨ מספר העמודות במטריצה</pre> <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
<pre>for (int i = 0 ; i < mat.GetLength (0) ; i++) mat [i, i] = ...</pre>	מטריצה ריבועית מסדר $n \times n$ אלכסון ראשי
<pre>int col = mat.GetLength(0) - 1; for (int i = 0 ; i < mat.GetLength (0) ; i++) { mat [i, j] = ... col -- ; }</pre>	אלכסון משני

string str ;	מחרוזת
str = "bla-bla-bla" ; str = new string ("bla-bla-bla") ;	מחרוזת ב-C# היא אובייקט
string str = Console.ReadLine();	קלט
פעולות שאינן משנות את המחרוזת	
num = str.Length;	אורך המחרוזת
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	חיפוש המופע הבא :
int place = str.IndexOf (subStr, fromPlace);	
int place = str.LastIndexOf (ch); // חיפוש מהסוף	חיפוש מהסוף :
int place = str.LastIndexOf (subStr);	
int place = str.LastIndexOf (ch, from); // חיפוש מהסוף ממקום	
int place = str.LastIndexOf (subStr, from);	