

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

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

int a, b; double x, y;	הגדרת משתנים :
---	-----------------------

מבנה התכנית בשפת C#:

<pre> public class שם-המחלקה { public static void Main(string[] args) { הגדרת המשתנים; הוראה; הוראה; } } </pre>	שים לב להזחות identification (קפיצת tab של תחילת השורה)
--	--

הערות

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

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

•

קלט/פלט:

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

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

הוראות השמה:

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

פעולות חשבוניות:

int a, b, c; double x, y, z;	נתונים המשתנים הבאים :
c = a + b;	חיבור
c = a - b;	חיסור
c = a * b;	כפל
שלם / שלם ← שלם	חילוק :

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:

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

בלוק הוראות:

	<pre>{ בלוק הוראות }</pre>	
--	--------------------------------	--

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;	if (פסוק-לוגי) ביצוע1;	אם ... אחרת ...

<pre>else b = b - 1;</pre>	<pre>else ביצוע; 2</pre>	(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>	אם... אחרת... (בלוק הוראות)
<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>	ברירת החלטה <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 (i = 1 ; i <= 5 ; i++) { Console.Write ("a number → "); a = int.Parse(Console.ReadLine()); sum = sum + a; }</pre>	<p>לולאת for</p> <p>(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</p> <p>(לולאה יורדת)</p>

while (ביטוי-לוגי) <pre>{ הוראה; הוראה; }</pre> <p>-----</p> <pre>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>	לולאת while במחברת הבחינה מותר לכתוב מחרוזות בעברית, אלא אם כן נאמר אחרת
--	--

פונקציות מתמטיות:

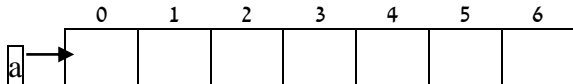
דוגמא	תחביר	מתקבל	מוחזר	טיפוס הערך
int b = Math.Abs(a); double y = Math.Abs(x);	Math.Abs(x)	שלם ממשי	שלם ממשי	ערך מוחלט $ x $
if (Math.Sqrt (x) > y) ...	Math.Sqrt (x)	שלם ממשי	ממשי	שורש ריבועי \sqrt{x}
double p = Math.Pow (x, b);	Math.Pow (x, y)	שלם ממשי	ממשי	חזקה x^y
double x = Math.Round(7.853)); int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

מספר אקראי - שימוש באובייקט Random

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

פעולות

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

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

מערך דו-ממדי (מטריצה)	<code>[5,7]</code> טיפוס-נתונים <code>arr = new [,]</code> טיפוס-נתונים
	<code>int [,] mat = new int [6,7] ;</code> <code>double [,] x = new double [N,M];</code>
פניה לתא	<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>
סריקת המערך	<p> <code>mat.GetLength (0)</code> \Rightarrow מספר השורות במטריצה <code>mat.GetLength (1)</code> \Rightarrow מספר העמודות במטריצה </p> <code>for (int i = 0 ; i < mat.GetLength (0) ; i ++)</code> <code> for (int j = 0 ; j < mat.GetLength (1) ; j ++)</code> <code> mat [i,j] =</code>

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 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 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);	