

עזרים C#

טווח מספרים	מה הוא מקבל	סוג המשתנה (טיפוס - type)
-/+ 2,147,483,647	מספר שלם (חיובי ושלילי)	int
0 - 4,294,967,295	מספר שלם (ללא סימן)	uint
עד 6-7 ספרות	מספר עם נקודה צפה (1.5)	float
עד 15 ספרות	מספר עם נקודה צפה (1.5)	double
-/+ 9,223,372,036,854,775,808	מספר שלם ארוך (חיובי ושלילי)	long
0 - 18,446,744,073,709,551,615	מספר שלם ארוך (ללא סימן)	ulong
-/+ 32,767	מספר שלם קצר (חיובי ושלילי)	short
0 - 32,768	מספר שלם קצר (ללא סימן)	ushort
	מחרוזת (נקרא גם מערך של תווים)	string
	תו (לדוגמא 'A')	char

פקודה להדפסה בקונסול:

```
Console.WriteLine("Hello World!");
```

פקודה לקליטת מחרוזת בקונסול:

```
string input = Console.ReadLine();
```

פקודה לקליטת מספר בקונסול:

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

סיינטקס לולאה בסיסית (לולאה שתרוץ 10 פעמים ותדפיס את המיקום):

```
for(int i=0; i<10; i++)
{
    Console.WriteLine(i);
}
```

סיינטקס של פונקציה:

```
static void func()
{
    Console.WriteLine("I am a function!");
}
```

ASCII תווים להדפסה			
32	space	64	@
33	!	65	A
34	"	66	B
35	#	67	C
36	\$	68	D
37	%	69	E
38	&	70	F
39	'	71	G
40	(72	H
41)	73	I
42	*	74	J
43	+	75	K
44	,	76	L
45	-	77	M
46	.	78	N
47	/	79	O
48	0	80	P
49	1	81	Q
50	2	82	R
51	3	83	S
52	4	84	T
53	5	85	U
54	6	86	V
55	7	87	W
56	8	88	X
57	9	89	Y
58	:	90	Z
59	;	91	[
60	<	92	\
61	=	93]
62	>	94	^
63	?	95	_
		96	`
		97	a
		98	b
		99	c
		100	d
		101	e
		102	f
		103	g
		104	h
		105	i
		106	j
		107	k
		108	l
		109	m
		110	n
		111	o
		112	p
		113	q
		114	r
		115	s
		116	t
		117	u
		118	v
		119	w
		120	x
		121	y
		122	z
		123	{
		124	
		125	}
		126	~

מערכים

מערך הוא אוסף סדור של תאים בזיכרון מטיפוס מסויים, שכל תא מכיל ערך מאותו הסוג. כשאנחנו רוצים להגדיר מערך, נעשה זאת בצורה הבאה:

```
int[] numbersArray = { 1, 5, 7, 8, 9 };
```

הדוגמא למעלה תיצור לנו מערך בגודל 5 תאים, וכל איבר במערך יהיה מסוג int. כדי שנוכל להבין טוב יותר איך זה נראה בפועל, נצפה בטבלה למטה:

0	1	2	3	4	מיקום (אינדקס)
1	5	7	8	9	איבר (ערך)

ניתן לקבל את גודל המערך גם אם אנחנו לא יודעים אותו מראש. נוכל לעשות זאת ע"י הפקודה השמורה הבאה:

```
Console.WriteLine(numbersArray.Length);
```

במקרה הזה הפלט שאקבל יהיה 5, בגלל שלמערך שלי יש חמש תאים.

כמו שאנחנו רואים בטבלה, מערכים מתחילים מהמיקום ה-0. בדרך כלל נרצה להשתמש בלולאה ע"מ לרוץ על איברי המערך, מכיוון שבעזרת לולאה אנחנו בכל פעם נשנה את האינדקס שלנו (לדוגמא: נעלה את האינדקס ב-1) ובכך נגיע לאיבר הבא במערך. דוגמא להדפסת כל איברי המערך:

```
for (int i = 0; i < numbersArray.Length; i++)
{
    Console.WriteLine(numbersArray[i]);
}
```

הלולאה תדפיס לנו את כל איברי המערך, אחד בכל שורה.

טבלאות אמת

טבלת אמת מאפשרת לנו לעקוב אחרי הקוד שלנו (בדרך כלל באותו קוד תהיה לולאה כלשהי ונרצה לעקוב אחר ההתקדמות שלה, ביחס לאינדקס – למיקום שלנו). לדוגמא ניקח את הקוד הבא:

```
string myString = "Hello World";
int x = 0;
for (int i = 0; i < myString.Length; i++)
{
    if(myString[i] == 'o')
        x++;
}
```

שורה	משתנה i	משתנה x	תנאי	פעולה (התנאי מתקיים?)
1	0	0	myString[0]'H'=='o'	לא
2	1	0	myString[1]'e'=='o'	לא
3	2	0	myString[2]'l'=='o'	לא
4	3	0	myString[3]'l'=='o'	לא
5	4	0	myString[4]'o'=='o'	x(0)+1
6	5	1	myString[5]' '=='o'	לא
7	6	1	myString[6]'W'=='o'	לא
8	7	1	myString[7]'o'=='o'	x(1)+1
9	8	2	myString[8]'r'=='o'	לא
10	9	2	myString[9]'l'=='o'	לא
11	10	2	myString[10]'d'=='o'	לא

בעזרת טבלת האמת, אנחנו עוברים שורה שורה בלולאה (כל פעולה בלולאה נקראת גם איטרציה). בצורה זו, אנחנו מבטיחים מעקב אחרי כל המשתנים שאנחנו רוצים לעקוב אחריהם. השורה האחרונה בטבלת המעקב (המסומנת באדום) אמורה להציג לנו את התוצאה הסופית של המשתנים. בדרך כלל במבחן יישאלו שאלות על המשתנים הסופיים ומה קורה איתם בסוף התרגיל. לכן עדיף להוסיף לטבלה כל משתנה שנרצה לעקוב אחריו.