

בסיס לתוכנה

```
static void Main()
{
    קוד
}
```

פלט | קלט

```
Console.WriteLine("text: " +
varText);
e.ReadLine();
```

רשם משיג קלט מהמשתמש כמחרוזת

text: (varText תוכן משתנה)

ניתן לכתוב את קוד זה בצורה הבאה:

```
var type.Parse(Console.ReadLine());
```

שמות משתנים

חוקים	מה לא לעשות
שמות לא מתחילים עם מספר	int 1number;
גם אם זה השם היחיד	
ללא סימנים חוץ מ_	int nice-1;
ללא רווחים	int nice 1;
אותיות גדולות וקטנות משנות בשם של המשתנה	

טיפים ללולאות פור

אם לולאה מתחילה באפס ויש לה תנאי של קטן מאינסוף, הולאה תרוץ אינסוף פעמים

```
for(int i= 0; i<10;i++)
```

לדוגמה הולאה הזו תרוץ עשר פעמים

אם רוצים שלולאה פור תרוץ פקודות שונות משתמשים בבלוק של פקודות

```
for(int i= 0; i<10;i++)
{
    קוד
}
```

לולאות פור יכולות להיות בתוך לולאות נוספות, קוראים לזה לולאות מקוננות

משתמשים בלולאות פור כאשר יודעים את כמות הפעמים שרוצים שהלולאה תרוץ

טיפים ללולאות וייל

צריך להשפיע על התנאי בתוך הלולאה אחרת תיווצר לולאה אינסופית

אין להשתמש בברייק בלולאות באופן כללי בגלל מגבלות בבגרות

שם התנאי הוא זקיף

דוגמאות למחלקה המתמטית

```
double num = -9.8; num = Math.Abs(-num); //9.8
int num = -9; num = Math.Abs(num); //9
double num1 = Math.Pow(2, 4); //16
double num1 = Math.Sqrt(25); //5
double num1 = Math.Round(25.9); //26
double num = Math.Max(2, 6); //6
double num = Math.Min(2, 6); //2
```

משפטי תנאי

מבנה של משפט תנאי if (תנאי)

סימן בין תנאים שמסמל צורך בכל התנאים &&

סימן בין תנאים שמסמל צורך באחד מן התנאים ||

ניתן להשתמש בסוגריים בתנאים בשביל || להשתמש גם ב&& וגם

התנאי | סימן | משמעות הסימן

x < y	<	x קטן מ y
x <= y	<=	x קטן או שווה ל y
x > y	>	x גדול מ y
x >= y	>=	x גדול או שווה ל y
x != y	!=	x אינו שווה ל y
x == y	==	x שווה ל y

ניתן להשתמש בערכים בוליאניים ובמשפטי תנאי

מסוג זה במשפט תנאי

בנוסף לכך, ניתן להשתמש בתנאים אלו גם בלולאות במקום של התנאים

for לולאות

(משפט 2; תנאי; משפט 1) for

משפט 1- רץ פעם אחת בתחילת הלולאה

משפט 2- רץ לאחר הבדיקה

תנאי- בדיקה שרצה בסוף הלולאה ואם התנאי נכון אז הלולאה חוזרת על עצמה

בלוקים של פקודות והזחות

בלוקים של פקודות בסי שארפ מתחילים ב { ונגמרים ב } . משתמשים בבלוק של פקודות על מנת לתחום פקודות בתוך משפט תנאי או לולאה הזחות הם המרווחים שמאפשרים לנו לראות ביותר נוחות את הקוד ולהבין אותו יותר בקלות.

while לולאות

משתמשים בלולאה הזו ברגע שכמות הפעמים שרוצים לבצע אותה אינה ידועה מבנה:
while(תנאי)
{
קוד
}

הגזלת מספר אקראי

פעם אחת רושמים:
Random rnd = new Random();
גבול עליון(אינו כולל), ולאחר מכן
rnd.Next((, (גבול תחתון(כולל));
אם רוצים מספר אקראי מאפס ועד לגבול):
מסויים) לא כולל
rnd.Next((אינו כולל));

Declaration | Cast | Usage

int (int) מספרים שלמים,
varName; במקרה של קאסט
הופך למספר שלם או
קוד אסקי

double (double) מספרים עשרוניים,
varName; מספרים ממשיים

char (char) אות יחידה מהמקלדת,
varName; הקאסט הופך מקוד
אסקי לאות

Declaration | Cast | Usage (cont)

string אין מחרוזת של אותיות
varName; ומספרים

bool (bool) True or False
varName;

const אם מוסיפים בהתחלה
varType לא של אז המשתנה לא
CAPITA- מתאים
LVA- const ישתנה
RNAME; לאורך התוכנית

שימוש במשתנים

Command	Usage
varName = 5	שנינו ערך משתנה קיים, int. המשתנה בדוגמה הוא
varType var1, var2, var3;	אפשר להגדיר כמה משתנים מאותו הסוג באותה שורה
varName ++/--;	מוסיף או מוריד אחד, עושה את varName = varName +/- 1

מתמטיקה

+	חיבור
-	חיסור
*	כפל
/	חילוק
%	שארית

מחלקה מתמטית

שם	פעולת הפונקציה	טיפוס שנשלח	טיפוס התוצאה המוחזרת
Math.Abs(num)	ערך מוחלט	double int	double int
Math.Pow(- מספר, מספר)	העלאת מספר בחזקה	double int	double
Math.Sqrt(- מספר)	שורש	double int	double
Math.Round(- מספר)	עיגול כלפי השלם הקרוב	double	-
Math.Max(- מספר, מספר)	מציאת המספר הגדול מבין שני המספרים	double	-
Math.Min(- מספר, מספר)	מציאת המספר הקטן מבין שני המספרים	double	-