

בסיס לתוכנה

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

פלט | קלט

| | |
|---------------------|---|
| Console.ReadLine(); | Console.WriteLine("-text: " + varText); |
| משיג קלט | רושם |
| מהמשתמש | תוכן משתנה) text: |
| כמחרוזת | varText) |

ניתן `vartype.Parse(Console.ReadLine())` לעשות בכל סוג משתנה שהוא לא מחרוזת

שמות משתנים

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

טיפים לולאות פור

אם לולאה מתחילה באפס ויש לה תנאי של קטן מאיקס, הלולאה תרוץ איקס פעמים

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

לדוגמה הלולאה הזו תרוץ עשר פעמים

אם רוצים שלולאת 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 לולאות 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); 4! התרגיל יהיה 2 בחזקת
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

| | |
|-----------|--|
| if (תנאי) | מבנה של משפט 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 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 | == | x == y |

ניתן להשתמש בערכים בוליאניים ובמשתנים מסוג זה במשפט תנאי

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

לולאות for

(משפט2; תנאי; משפט1)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
}

הגרלת 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((גבול עליון(אינו כולל));

Declaration | Cast | Usage

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

double (double) מספרים עשרוניים, מספרים ממשיים
varName;

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

Declaration | Cast | Usage (cont)

string אין מחרוזת של אותיות ומספרים
varName;

bool (bool) True or False
varName;

const לא אם מוסיפים בהתחלה של אז const המשתנה לא יהיה ניתן לשינוי
varType מתאים
CAPITA-
LVA-
RNAME;

שימוש במשתנים

Comm- Usage and

varName .int שינוי ערך משתנה קיים, int = 5 המשתנה בדוגמה הוא

varType אפשר להגדיר כמה משתנים מאותו הסוג באותה שורה
var1,
var2,
var3;

varName מוסיף או מוריד אחד, עושה את varName = ++/--; אותו הדבר כמו varName +/- 1

מתמטיקה

+ חיבור

- חיסור

* כפל

/ חילוק

% שארית

מחלקה מתמטית

| שם | פעולת הפונקציה | טיפוס שנשלח | טיפוס התוצאה המוחזרת |
|----------------------|------------------------------------|-------------|----------------------|
| Math.Abs(num) | ערך מוחלט | double | double |
| Math.Pow(מספר, מספר) | העלאת מספר בחזקה | double | double |
| Math.Sqrt(מספר) | שורש | double | 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 | - |



By The Guy From Python (sgfggrfgfdsdf)

Published 2nd November, 2020.

Last updated 8th March, 2021.

Page 2 of 2.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>