

## פעולות (או פונקציות או מתודות או שיטות)

בשפה מונחית עצמים, כמו C# מבחינים בין שני סוגי פעולות: 1. סטאטית 2. דינמית (שאינה סטאטית)

### סטאטית

פעולה סטאטית נקראת גם 'חופשית' או 'עצמאית' מהסיבה שאינה קשורה לעצם מסוים, אלא היא כללית וניתנת לקריאה מכל מקום ובכל הקשר.

פעולה היא כמו מיני-תוכנית. גם ה-Main שלנו, שם כתבנו את הפקודות עד היום, הוא בעצם פעולה.

היכן מגדירים פעולה סטאטית?

בינתיים נגדיר אותה באותה מחלקה שבה מוגדרת ה-Main או לפני או אחרי ה-Main לדוגמה:

```
using System;
```

```
namespace myprogram1  
{  
    class MainClass  
    {
```

כאן נגדיר פעולה סטאטית

```
        public static void Main(string[] args)  
        {  
            Console.WriteLine("Hello World");  
        }  
    }  
}
```

### כעת נראה את המרכיבים של הפעולה

1. כללי נגישות לפעולה (כמעט תמיד public) - על מנת לקרוא לפעולה מכל מקום בתוכנית היא צריכה להיות מוגדרת עם public.

2. המילה: static

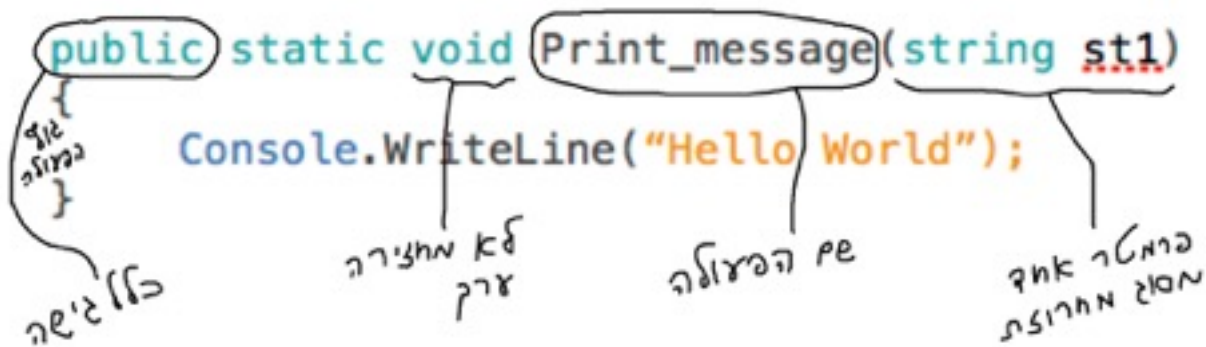
3. סוג ערך מוחזר מהפעולה - אם הפעולה לא מחזירה ערך, יכתב: void

4. שם הפעולה - אותם כללים כמו שם של משתנה.

5. סוגרים ובתוכם יוגדרו משתנים שלהם נקרא פרמטרים

6. גוף הפעולה כולו מוכל בין סוגרים מסולסלים.

(המרכיבים מ-1 ועד 5 ביחד מהווים את 'כותרת הפעולה')



גוף הפעולה מכיל פקודות רגילות של C#, כמו שכתבנו עד היום ב-Main. הפרמטרים הם כמו הקלט שהיה לנו ב-Main. אלה הם ערכים שינתנו לפעולה כאשר נקרא לה.

אם סוג הערך המוחזר אינו void (כל דבר אחר כגון: bool, string, int וכדומה), תהיה בתוך הפעולה פקודת: **ערך return**

נתקלנו כבר במצבים שכאשר הפעולה הוגדרה עם סוג ערך מוחזר והייתה אפשרות להגיע לסיום הפעולה ללא הגעה ל- return - קיבלנו הודעת שגיאה.

אם במהלך ביצוע הפעולה, הגענו לפקודת return, באותו רגע הסתיימה הפעולה, והשליטה חוזרת למקום שממנו קראנו לפעולה. אם זו פעולת void היא תסתיים בהגיענו לסוגר המסולסל האחרון של גוף הלולאה.

דוגמה לפעולה שמקבלת 2 פרמטרים, הראשון מסוג int והשני מסוג string ומחזירה ערך מסוג .string

```
public static string calculate(int num, string str1)
{
    int k;
    if (num > 3)
    {
        k = num*2;
    }
    else
    {
        k=num*3;
    }
    string message = str1+ k;
    return message;
}
```

הסיבה שהגדרנו פעולה, היא כדי שנוכל להשתמש בה, אז כעת נראה כיצד להשתמש בה.

```
public static void Main(string[] args)
{
    int r=5;
    string s1= "Here is the result: ";
    Console.WriteLine(calculate(r,s1)

}
```

הפעלנו את הפעולה: calculate בתוך פקודת ההדפסה. הפרמטרים שסיפקנו לפעולה, מוגדרים ב-Main ושם גם ניתן להם ערך. שימו לב שהשמות של הפרמטרים שהגדרנו בפעולה אינם זהים לשמות הפרמטרים שאיתם קראנו לפעולה.

צריכה להיות התאמה בסוג ובסדר של הגדרת הפרמטרים בפעולה, ונתינתם בקריאה לפעולה. בדוגמה: r הוגדר כ-int והוא תואם את: num בהגדרה, s1 הוגדר כ-string והוא תואם את: str1 בהגדרת הפעולה.

לא ניתן להשתמש בפעולה במשתנים שמוגדרים ב-Main ולהיפך. הפעולה היא יחידת תכנות עצמאית שאפשר רק להפעיל מה-Main. המשתנים שמוגדרים בתוך הפעולה נקראים 'מקומיים' והם 'נעלמים' כשאנו יוצאים מהפעולה.

הרעיון המרכזי מאחורי פעולות זה לפשט את התוכנית ולעשותה מה שנקרא: מודולארי. פעולה גם יכולה להיקרא ממספר מקומות, על ערכים שונים בכל פעם.

פעולה יכולה לקרוא לפעולה אחרת.

## תרגילים

כתבו פעולה בשם swap שלא מחזירה ערך. היא תקבל מה-Main שני מספרים שלמים במשתנים, למשל num1 ו-num2 ותחליף את ערכם. ה-Main תקרא את המספרים מהמשתמש.

הפכו את תוכנית ההסבה מצלזיוס לפרנהייט לפעולה. היא תקבל מספר מסוג int ותחזיר מספר מסוג double.

בתוכנית הראשית צרו לולאת while שתקלוט טמפרטורה בצלזיוס, תסב אותה בעזרת הפעולה, תדפיס גם את הטמפ' בצלזיוס וגם בפרנהייט, ותבקש שוב טמפרטורה בצלזיוס. אם הוקש המספר מינוס מאה, התוכנית תסתיים.