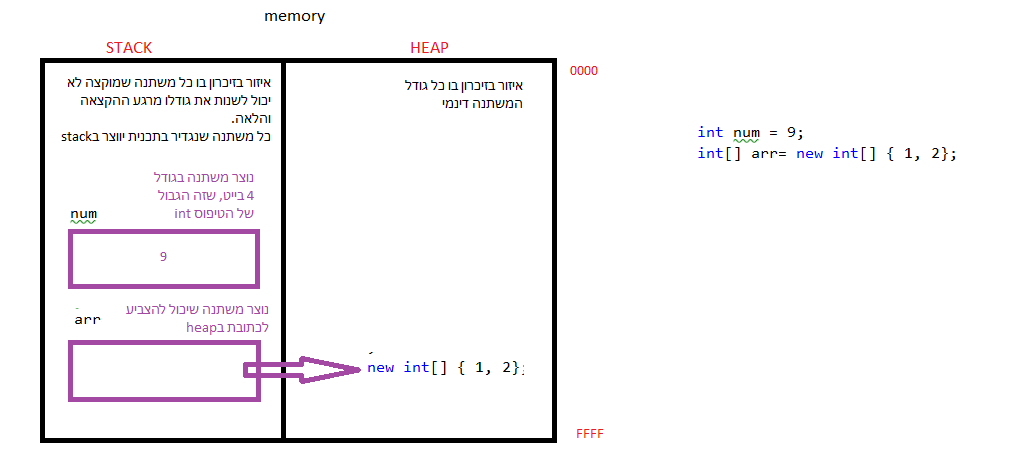
סיכום חומר ב- C#

**פלט קלט בסיסי**

* הכנסת קלט מהמשמש : str = Console.ReadLine();
  + `ReadLine()` - פונקציה המחזירה את הקלט מהמקלדת **כמחרוזת**.
  + זה לוקח את כל הקלט עד שהמשתמש לוחץ `להיכנס`.
* הדפסת פלט : Console.WriteLine("Hello world\n");
  + קיצור נעשה על ע''י : CW+TAB+TAB.
  + במידה ונרצה להדפיס משתנים בתוך ההדפסה נשתמש בסימן $ לפני הגרשיים(" "$).
  + אם נרצה להדפיס משתנים, פעולות מתמטיות וכ"ו, נרשום בתוך סוגרים מסולסלים בצורה הבאה : {Variable} , {1+1} .
  + דוגמא למימוש : Console.WriteLine($" msg is : {str} ");.

**סוגי טיפוסים בסיסיים** (רק מה שמשתמשים בדרך כלל, לא מופיע פה הכל)

* טיפוסים שבהם נשתמש בדרך כלל בשפה הם:
  + Int- מספרים שלמים, גודל 4 bytes .
  + Float -מספרים עשרוניים, גודל 4 bytes.
  + Double- מספרים עשרוניים, גודל 8 bytes
  + Decimal- מספרים עשרוניים, גודל 16 bytes, יותר מדויק מ – Float ו- Double.
  + Bool- בוליאני , יכיל בתוכו רק True או False.
  + Char- ערך תווי(תו בודד), גודל 2 bytes .
  + String- מכיל בתוכו מערך של תווים.
* Ref vs val- דיאגרמה בזיכרון, דוגמה להמחשה:
* המרת מחרוזת למספר & מספר למחרוזת:
  + המרה ב C# נעשית בדרך כלל על ידי השימוש במילה Convert.
  + באמצעות כך ניתן להמיר מחרוזת למספר וגם להפוך מספר למחרוזת.
  + דוגמאות להמרה מ string למספר וגם ממספר ל- string:

|  |
| --- |
| -------------------- Convert from string to number ----------------- |
|  |

|  |
| --- |
| int num1 = Convert.ToInt32("3"); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| double num2 = Convert.ToDouble("7.8"); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| -------------------- Convert from number to string ----------------- |
|  |

|  |
| --- |
| string strNum1 = num1.ToString(); |
|  |

|  |
| --- |
|  |
|  |

string strNum2 = num2.ToString();

* המרה מרומזת:
  + המרה מרומזת תתבצע כאשר נשים משתנה מטיפוס אחד למשתנה מטיפוס אחר כל עוד אין חשש שנאבד מידע במהלך ההמרה.
* המרה מפורשת:
  + המרה מפורשת תתבצע כאשר נשים משתנה מטיפוס אחד למשתנה מטיפוס אחר ויש חשש שנאבד מידע במהלך ההמרה.

**מערך ומטריקס:**

* מערכים:
* מערך ב-C# נעשה באמצעות סוגרים מרובעות "[]" אחרי קביעת הטיפוס ולפני שם המשתנה.
* ב-C# ישנם מספר דרכים ליצירת מערכים, בדרך כלל נשתמש בצורה הבאה:

1. דרך ראשונה ליצירת מערך-

באמצעות ציון מספר התאים שיהיו לאותו מערך ללא צורך לתת את ערכי התאים בשורת האתחול, דוגמה :

|  |
| --- |
| int[] arr1 = new int[3]; |
|  |

|  |
| --- |
| arr1[0] = 2; |
|  |

|  |
| --- |
| arr1[1] = 4; |
|  |

arr1[2] = 8;

1. דרך שנייה ליצירת מערך- באמצעות ציון מספר התאים שיהיו לאותו מערך והשמת ערכי התאים בשורת האתחול, דוגמה : int[] arr2 = new int[3] { 2, 4, 8 }; .
2. דרך שלישית ליצירת מערך-
   * באמצעות השמת ערכי התאים בשורת האתחול ללא צורך בציון מספר התאים שיהיו לאותו מערך.
   * מספר התאים יקבע לפי מספר הערכים שניתנו באתחול.

* מטריקס:
* מטריקס ב- C# נעשה באמצעות 2 סוגרים מרובעים "[][]" אחרי קביעת הטיפוס ולפני שם המשתנה.
* ב-C# ישנם מספר דרכים ליצירת מטריצות, בדרך כלל נשתמש בצורה הבאה:
* דרך ראשונה ליצירת מטריקס: int[][] matrix1 = {new int[]{ 1, 1, 1 }, new int[] { 1, 1, 1 }};.
* דרך שניה ליצירת מטריקס:

int[][] matrix2 = new int[2][] { new int[] { 1, 1, 1 }, new int[] { 1, 1, 1 } }; .

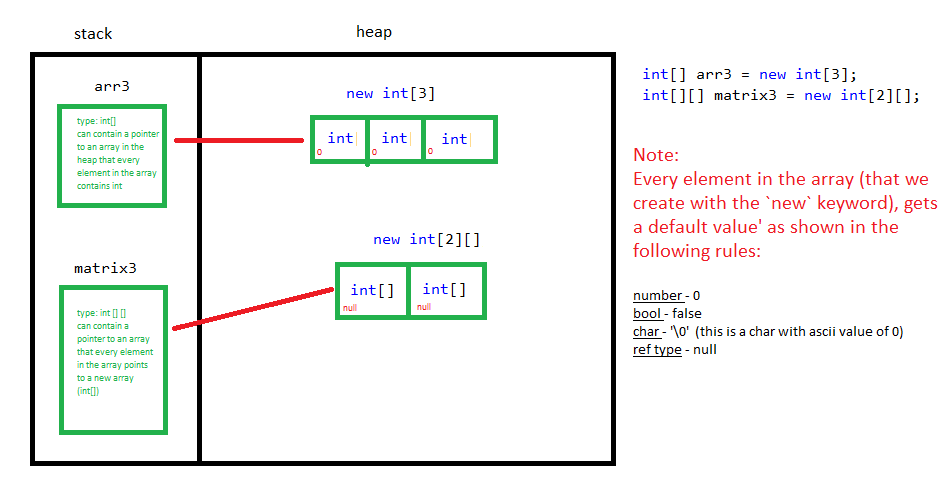
* דרך שלישית ליצירת מטריקס:

|  |
| --- |
| int[][] matrix3 = new int[2][]; |
|  |

|  |
| --- |
| matrix3[0] = new int[3] { 11, 12, 13 }; |
|  |

matrix3[1] = new int[3] { 21, 22, 23 };

* מערך VS מטריקס - תרשים בזיכרון:

****

**תנאים**

* שלוש אופציות לשימוש בתנאים:
  + אופציה ראשונה- עם משתנה בוליאני: כאשר נשתמש בטיפוס בוליאני אין צורך להשוותו לערך True,כי **בצורה דיפולטיבית** המשתנה נבדק האם הוא מכיל בתוכו True.
  + אופציה שנייה- עם ביטוי שמחזיר ערך בוליאני : כאשר נשתמש בביטוי (דוגמה כמו גדול או קטן) בתוך תנאי ייבדק האם זה True או False.
  + אופציה שלישית- עם ערך בוליאני ישיר(True/False).
* תנאים מקוננים-תנאי בתוך תנאי.
* תנאי מקוצר:
  + תנאי זה נעשה באמצעות שורה אחת בלבד.
  + ניתן לבדוק ביטוי או משתנה שמכיל בתוכו ערך בוליאני באמצעות שימוש בתנאי מקוצר.
  + התוצאה המתקבלת תיכנס לתוך משתנה.
  + רישום תנאי מקוצר נעשה בצורה הבאה:

Bool flag = true;

String res;

res=(flag)?"flag is true" : "flag is false";

if =true? true else

Console.WirteLine($"res:{res}");

**פונקציות**

* פונקציה שאינה מקבלת פרמטרים ואינה מחזירה ערך.
* פונקציה שמקבלת פרמטרים ואינה מחזירה ערך.
* פונקציה שאינה מקבלת פרמטרים אך מחזירה ערך.
* פונקציה במקבלת פרמטרים ועושה ערך מוחזר.