

סיכום שיעור היום דיברנו על משפט התנאי

- תנאי שעל פיו המחשב יבצע סט מסויים של פקודות אם התנאי מתקיים, או סט אחר במידה והתנאי לא מתקיים. כך עובדת פקודת התנאי (if)

```
if (condition)
{
    פקודות שתתבצענה אם התנאי נכון
}
else
{
    פקודות שתתבצענה אם התנאי אינו נכון
}
```

אמרנו גם שיכול להיות if ללא else, אבל לא להיפך.

התנאי שנמצא בסוגריים זהו ביטוי שיכול להיות נכון או לא נכון. ראינו דוגמה לביטוי אריתמטי, כמו:  $a > b$  או השוואה:  $a == b$  (נא לזכור שבתנאי שני סימני שווה זו למעשה שאלה - האם שני הצדדים שווים), בניגוד לסימן שווה אחד שמציין השמה של מה שמימין למה שמשמאל לסימן.

הפקודות בבלוק שמיידית אחרי ה- if שייכות אליו. ללא הסוגריים המסולסלות, המחשב ישייך רק פקודה עוקב אחת ל- if.

אמרנו גם שפקודות התנאי יכולות לקונן, כלומר להיות אחת בתוך השניה, ואז משייכים את הסוגר } לפותח המתאים, על פי המיקום של התנאי.

**ראינו דוגמה לבדיקת מי המספר משלושה, הגדול ביותר.**

```
//We received 3 numbers from the input into variables: a,b,c
if (a>b) //if number 1
{
    if (a>c) //if number 2
    {
        Console.WriteLine(a); //if we got here: a>b and a>b
    }
    else //else of if number 2
    {
        Console.WriteLine(c); //if we got here: a>b and c>=a
    }
}
else //else of if number 1
{
    if(b>c) //if number 3
    {
        Console.WriteLine(b); //if we got here: a<=b and b>c
    }
    else //else of if number 3
    {
        Console.WriteLine(c); //if we got here: b>=a and c>=b
    }
}
```

אם שמתם לב, העימוד של התוכנית הנ"ל נעשה כך שלכל פותח מסולסל, יש סוגר באותה עמודה מתחתיו (תיתכנה הרבה שורות מתחת). הייתי מציע בכל פעם שיש לכתוב בלוק חדש, לכתוב את הפותח והסוגר אחד מתחת לשני, ואז לתת את הפקודות ביניהם.

דוגמה למצב מטעה:

```
int a=2, b=3, c=4;
if (a>b)
    a=b;
    c=b;
Console.WriteLine(a,b,c);
```

מה יודפס? אם אמרתם: 3 3 3 טעיתם. מה שיודפס זה: 2 3 3 הפקודה a=b לא מתבצעת כיוון שמלכתחילה b>a, אבל הפקודה c=b מתבצעת תמיד ללא תלות ב- if. זה למרות שעל פי העימוד זה נראה שגם c=b "שיכת" ל- if.

### חזרה על מה הם משתנים בתוכנית

**הגדרנו שטח בזיכרון התוכנית** (אם תזכרו, התוכנית רצה בזיכרון הפנימי של המחשב - Ram). והשטח היה למשל מסוג: **string**. זהו סוג שיכול לקבל תוים כלשהם מתוחמים בין צמד גרשיים. האוסף הזה של תוים ניקרא: מחרוזת (זה פירוש המילה string באנגלית).

**לשטח הזה קראנו 'משתנה'** - זוהי הגדרה מאד שימושית בתכנות וכמעט בכל תוכנית אנחנו נגדיר משתנים. הסיבה שהם ניקראים משתנים, זה כי השטח בזיכרון יכול להשתנות.

הצורה שבה הגדרנו את המשתנה הייתה כך:

```
string name = "Siri";
```

המילה string היא סוג המשתנה - כאמור המשתנה יכול להכיל ערך מסוג מסוים, במקרה זה הסוג הוא מחרוזת. השם שנתנו למשתנה הוא: name. זוהי הצורה הכללית להגדרת משתנים:

; שם-משתנה סוג-משתנה

בנוסף להגדרה, גם עשינו השמת ערך למשתנה, על ידי סימן השווה ולכן מה שעשינו כללית היה:

; ערך = שם-משתנה סוג-משתנה

(לא לשכוח את תו הנקודה פסיק בסוף)

**אז אם נתרגם למילים את הפקודה הזו, ניכתוב:**

הגדר משתנה בשם name שיכול להכיל מחרוזת (אוסף תוים), והכנס אליו את הערך של שמך בין צמד של גרשיים.

התכונה החשובה של משתנה זה שהוא יכול להשתנות

לדוגמה, אם היינו כותבים את הקטע הבא:

```
string n1= "nice";
n1="day";
Console.WriteLine(n1);
```

מה יודפס? אם אמרתם: day אז צדקתם. ( למדנו גם לעקוב אחרי ערך המשתנים בטבלה)

קודם הכנסנו למשתנה n1 (שהוא מסוג מחרוזת) את הערך "nice", שורה אחרי זה הכנסנו לאותו משתנה n1 את הערך "day" אז הערך האחרון ששמנו בו, הוא הקובע, ולכן כשהדפסנו אותו עם ה- WriteLine, קיבלנו: day

איך קובעים שם של משתנה? זהו בעיקרון צירוף של אותיות ומספרים שמתחיל באות (ההגדרה הרשמית קצת יותר מורכבת, אז בינתיים נסתפק בזה). אסור להגדיר משתנה בשם של מילה שמורה בשפת C#, מילים שמורות הן למשל: int, if, string (יש עדין כאלה שלא למדנו).

דוגמה לשמות חוקיים של משתנים:

```
a
x
name3
lastname
```

דוגמה לשמות לא חוקיים למשתנים

```
123    משתנה לא יכול להכיל רק מספרים
4abc   משתנה לא יכול להתחיל במספר.
aa bb  שם משתנה לא יכו לכללור רווח.
else   זוהי מילת מפתח
```

נדבר בהמשך עוד רבות על משתנים, ונלמד סוגים שונים של משתנים. לסיכום: משתנה הוא שם של שטח בזיכרון בו ניתן לשמור ערכים מסוג מסויים (נקרא סוג המשתנה). שם המשתנה מורכב מאותיות וספרות ומתחיל באות.

זה לא חובה, אבל כדאי להגדיר שמות משתנים בעלי משמעות, כדי שמי שקורא את התוכנית (כולל מי שכתב אותה), יהיה לו מושג בערך מה היא עושה.

למשל בשיעור קראנו למשתנה שהכיל את השם: name ולכן קל לזכור מה המשמעות שלו. אם הייתם קוראים לו x, זה היה חוקי אבל פחות מובן.

**שפת C# מבחינה בין אות קטנה לאות גדולה** (באנגלית), כך ש: Name ו-name הם שני משתנים שונים לגמרי.

שיעורי בית  
לכתוב תוכנית שקולטת 3 מספרים ומדפיסה את שלושתם מהקטן לגדול.