BasicIfStatements

Created time	July 13, 2026 5:09 AM
Category	Conditionals
Last edited by	R.A R.A
Last updated time	August 12, 2026 5:20 AM

```
bool isComplete = true;

if (isComplete)
{
    Console.WriteLine("The statement was true.");
    Console.WriteLine("This line works in true.");
}
else
{
    Console.WriteLine("The statement was false.");
    Console.WriteLine("This should also run");
}



//~~~
```

```
Console.Write("What is your first name: ");
string? firstName = Console.ReadLine();

string lastName;

if (firstName.ToLower() == "tim")
{
    Console.WriteLine("Hello Mr. Corey.");
    lastName = "Corey";
}
else
{
    Console.WriteLine($"Helo {firstName}");
    lastName = "Smith";
}

Console.WriteLine(lastName);
Console.WriteLine("End of program.");
```

משפטי if הם חלק בסיסי מאוד מהאופן שבו אנחנו עובדים עם C#.

תנאי מאפשר לנו ליצור לוגיקה שמתפצלת לענפים שונים.

נתחיל עם bool.

נאתחל אותו לערך ההתחלתי שלו, כברירת מחדל - **false**.

קודם כול יש לנו **if**.

אחריו יש לנו זוג סוגריים עגולים **()**, וכל מה שנמצא בתוך ה-**()** ייבדק כדי לראות אם הוא **true**.

```
if ()
{

}
```

ויש לנו סוגר מסולסל פותח וסוגר מסולסל סוגר **{ }**, וביניהם יהיה הקוד שירוץ אם מה שנמצא בתוך ה-**()** הוא **true**.

יש לנו **bool**, וה-**bool** הוא **true** או **false**.

אם נכתוב:

```
if (isComplete)
{

}
```

זה יספק את משפט ה-**if** שלנו.

מפני שהוא יסתכל על ה-**bool** ויבדוק אם ה-**bool** הוא **true** או **false**.

אנחנו יכולים לכתוב:

```
if (isComplete)
{
    Console.WriteLine("המשפט היה אמת.");
}
```

בתחתית הקוד אנחנו כותבים:

```
Console.WriteLine("סוף התוכנית.");
```

כך:

```
if (isComplete)
{
    Console.WriteLine("המשפט היה אמת.");
}

Console.WriteLine("סוף התוכנית.");
```

וזה חשוב מפני שכרגע

```
(isComplete)
```

הוא **false**.

כלומר, אנחנו לא אמורים להריץ את הקוד הזה.

אם נגדיר אותו ל-**true**:

```
bool isComplete = true;
```

הקוד שלנו ירוץ, ואנחנו נראה את:

```
Console.WriteLine("המשפט היה אמת.");
```

```
bool isComplete = true;

if (isComplete)
{
    Console.WriteLine("המשפט היה אמת.");
}

Console.WriteLine("סוף התוכנית.");
```

כלומר, לפי השאלה האם הביטוי/הערך הזה הוא **true** או **false**:

```
(isComplete)
```

נקבע האם אנחנו הולכים להריץ את הקוד הזה:

```
{
    Console.WriteLine("המשפט היה אמת.");
}
```

משפט else

```
else
```

פשוט אומר: אחרת, וגם לו יש סוגר מסולסל פותח וסוגר מסולסל סוגר **{ }**.

ואנחנו יכולים לכתוב:

```
Console.WriteLine("המשפט היה שקר.");
```

רק כדי להראות לנו איזה קוד רץ.

```
bool isComplete = true;

if (isComplete)
{
    Console.WriteLine("המשפט היה אמת.");
}
else
{
    Console.WriteLine("המשפט היה שקר.");
}

Console.WriteLine("סוף התוכנית.");
```

אז עכשיו יש לנו: אם זה **true**:

```
if (isComplete)
```

תריץ את זה:

```
{
    Console.WriteLine("המשפט היה אמת.");
}

else / אחרת, תריץ את זה:

{
    Console.WriteLine("המשפט היה שקר.");
}
```

כלומר, לעולם לא יהיה מצב שבו גם:

```
Console.WriteLine("המשפט היה אמת.");
```

וגם:

```
Console.WriteLine("המשפט היה שקר.");
```

ירוצו.

זה או האחד או השני.

אז אם נשנה את ה-**bool** ל-**false**:

```
bool isComplete = false;
```

עכשיו יודפס:

```
"המשפט היה שקר."
```

מפני שמשפט ה-**else** רץ.

זהו משפט **if**/**else** בסיסי.

לא היינו חייבים להשתמש במשפט **else**.

אנחנו יכולים להשתמש רק במשפט **if**.

אבל אנחנו לא יכולים להשתמש רק במשפט **else**.

מפני שהוא חייב להיות קשור למשהו כדי לדעת אם התנאי הוא **true** או **false**.

סימן השוויון הכפול **==** הוא פעולת השוואה שנמצאת בתוך ה-**()**,
והיא תחזיר לנו ערך בוליאני: **true** או **false**.

```
if (firstName == "Tim")
```

הוא משווה בין שני ערכים.

כלומר הוא בעצם שואל:

האם השם הפרטי שלו שווה ל-**Tim**.

וההשוואה תחזיר **Boolean**, כלומר **true** או **false**.

אנחנו צריכים לזכור שה-**()** האלה מחושבים לערך **Boolean**:

```
(firstName == "Tim")
```

מה קורה אם המשתמש כותב **tIM**:

```
Console.Write("מהו השם הפרטי שלך: ");
string? firstName = Console.ReadLine();

if (firstName == "Tim")
{
    Console.WriteLine("שלום מר קורי.");
}
else
{
    Console.WriteLine($"שלום {firstName}");
}

Console.WriteLine("סוף התוכנית.");
```

למה כאשר הערך היה **t** באות קטנה ו-**IM** באותיות גדולות, לא הודפס:

```
"שלום מר קורי."
```

הסיבה היא משפט ה-**if**:

```
if (firstName == "Tim")
```

הביטוי הזה מחזיר **false**.

אז למה הוא מחזיר **false**?

מפני שההשוואה רגישה לאותיות גדולות וקטנות

```
case sensitive
```

ולכן רק **T** גדולה ואחריה **im** קטנות תגרום ל-**Tim** להתאים ל-**Tim**.

האם זה באמת מה שאנחנו רוצים?

לא, מפני שזה לא פתרון כל כך עמיד.

יש מתודה של **string** שאפשר להשתמש בה:

```
.ToLower()

if (firstName.ToLower() == "tim")
```

היא ממירה את המחרוזת של השם הפרטי לאותיות קטנות.

עכשיו כבר לא משנה אם המשתמש כתב אותיות גדולות או קטנות.

חשוב לציין שהיא לא משנה את המשתנה המקורי.

אנחנו כנראה נשתמש ב:

```
.ToLower()
```

די הרבה כאשר נעבוד עם **strings** ונבצע השוואת מחרוזות כמו כאן:

```
if (firstName.ToLower() == "tim")
```

שימו לב, יכולים להיות לנו משפטי **if** מקוננים

```
nested if statements
```

```
Console.Write("מהו השם הפרטי שלך: ");
string? firstName = Console.ReadLine();

if (firstName.ToLower() == "tim")
{
    Console.WriteLine("שלום מר קורי.");
}
else
{
    Console.WriteLine($"שלום {firstName}");
}

Console.WriteLine("סוף התוכנית.");
```

אנחנו יכולים להכניס באזור הזה משפט **if** נוסף:

```
if (firstName.ToLower() == "tim")
{
    Console.WriteLine("שלום מר קורי.");
}
```

זה תלוי בנו אם אנחנו רוצים לעשות את זה.
זה אפשרי, ואפשר להכניס לתוך ה-**scope** משתנים ומשפטי **if**.

לדוגמה, ניצור משתנה עם ערך שכתוב ב- **hard coded**:

```
Console.Write("מהו השם הפרטי שלך: ");
string? firstName = Console.ReadLine();

if (firstName.ToLower() == "tim")
{
    Console.WriteLine("שלום מר קורי.");
    string lastName = "Corey";
}
else
{
    Console.WriteLine($"שלום {firstName}");
}

Console.WriteLine(lastName);
Console.WriteLine("סוף התוכנית.");
```

שימו לב, **IntelliSense** לא מציג לנו את המשתנה, ואם בכל זאת ננסה להקליד אותו, נקבל:

```
"The name 'lastName' does not exist in the current context"
```

כלומר:

```
"השם 'lastName' אינו קיים בהקשר הנוכחי.
```

הסיבה היא שהמשתנה הזה נוצר בתוך ה-**scope** הזה:

```
{
    Console.WriteLine("שלום מר קורי.");
```

```
    string lastName = "Corey";
}
```

כלומר, המשתנה לא יוצא מחוץ ל-**scope** הזה.

אנחנו יכולים להעביר דברים פנימה אל **scope**, אבל לא החוצה ממנו.

אם אנחנו רוצים שהמשתנה יהיה זמין גם מחוץ ל-**scope**, אנחנו צריכים להצהיר עליו למעלה, ואז נוכל לכתוב:

```
lastName = "Corey";
```

כך זה נראה:

```
Console.Write("מהו השם הפרטי שלך: ");
string? firstName = Console.ReadLine();

string lastName;

if (firstName.ToLower() == "tim")
{
    Console.WriteLine("שלום מר קורי.");
    lastName = "Corey";
}
else
{
    Console.WriteLine($"שלום {firstName}");
    lastName = "Smith";
}

Console.WriteLine(lastName);
Console.WriteLine("סוף התוכנית.");
```

עכשיו שינינו את המשתנה בתוך ה-**scope**, אבל מכיוון שהוא נוצר ב-**scope parent** , אנחנו יכולים לגשת אליו שוב מתוך ה-**scope** האב.