Comparison Operators

When writing a program, we often need to check if a value is correct or compare two values. Comparison operators allow us to compare values and evaluate their relationship. Rather than evaluating to an integer, they evaluate to boolean values. Expressions that evaluate to boolean values are known as boolean expressions.

Comparison operators include:

> *Equals* `==`: returns true if the value to the left is equal to the value to the right.
>
> *Inequality operator* `!=`: returns true if the two values are not equal.
>
> *Less than* `<`: returns true if the value to the left is less than the value to the right.
>
> *Greater than* `>`: returns true if the value to the left is more than the value to the right.
>
> *Less than or equal to* `<=`: returns true if the value to the left is less than or equal to the value on the right.
>
> *Greater than or equal to* `>=`: returns true if the value to the left is more than or equal to the value to the right.

Here's what a boolean expression using comparison operators can look like:

```
bool answer = 3 < 75;
Console.WriteLine(answer); // prints True
```

In this example, we use the less than `<` comparison operator to compare the values `3` and `75`, then save the expression to a variable named `answer` with a `bool` data type. If we were to print the value of `answer` to the console, it would print out `True`, since the number 3 is less than the number 75.

In addition to comparing integers, we can also compare variables, strings, and even boolean values:

```
bool answer = (true == false);
Console.WriteLine(answer); //prints False
```

Here, we use the equals comparison operator to see if the Boolean value `true` is equal to `false`. This time, the expression evaluates to false. We'll look more into comparing boolean values in the next exercise.

☑Instructions

1.

You are driving to your family's house for a holiday and want to see if you'll get there before dinner. Dinner will begin at 6:00 PM and the house is 95 miles away. If you leave at 2:00PM and drive an average of 30 miles per hour, will you get there early (before 6:00pm)?

Create a `double` variable named `timeToDinner` and save the difference in hours between 2:00pm and 6:00pm.

✓

Hint

The difference is 4 hours, so we can save the value `4` to the variable.

2.

Save the value `95` to a `double` variable named `distance`.

Save the value `30` to a `double` variable named `rate`.

✓

Hint

Use the `double` data type for these values:

```
double myNumber = 583;
```

3.

We can calculate how long it will take us by dividing the `distance` variable by the `rate` variable.

Write out the expression and save it to the variable `tripDuration`.

✓

Hint

Use the following forumla:

$$time = distance/rate;$$

4.

Create a `bool` variable named `answer`. Save the appropriate comparison that checks if `tripDuration` is *less than or equal to* `timeToDinner`.

✓

Hint

Since we know that `tripDuration` needs to be the same or less than `timeToDinner`, we should use a *less than or equals to*comparison operator:

```
bool result = (number <= otherNumber);
```

5.

Print `answer` to the console. Will you arrive before dinner begins?

✓

Hint

Use `Console.WriteLine()` to print something to the console:

```
Console.WriteLine("hello");
```