Built-In Methods

So how do we do more advanced mathematical operations? For example, how would we perform a square root on a number if the program doesn't recognize a square root symbol?

There are several built-in methods that we can use to manipulate numerical data and perform more complex mathematical calculations. Here are a few:

`Math.Abs()` —will find the absolute value of a number.
Example: `Math.Abs(-5)` returns 5.

`Math.Sqrt()` —will find the square root of a number.
Example: `Math.Sqrt(16)` returns 4.

`Math.Floor()` —will round the given double or decimal down to the nearest whole number. Example: `Math.Floor(8.65)` returns 8.

`Math.Min()` —returns the smaller of two numbers. Example: `Math.Min(39, 12)` returns 12.

☑Instructions

1.

In this exercise, we're going to use built-in methods to determine which number is smaller between the square roots of two different numbers.

First, find the square root of `numberOne` and round the answer down so it doesn't have a decimal. Save this value to a new `double` variable `numberOneSqrt`.

Hint

We can chain methods by putting methods inside of each other:

```
Math.Floor(Math.Sqrt(number));
```

2.

Do the same process to variable `numberTwo` and save this value to a new `double` variable `numberTwoSqrt`.

Hint

Make sure when you are chaining methods that you have the correct number of parentheses:

```
Math.Floor(Math.Sqrt(number));
```

3.

Inside of a `Console.WriteLine()` statement, use a built-in method that returns the smallest of two numbers, using the values `numberOneSqrt` and `numberTwoSqrt`.

Which value gets printed to the console?

Hint

Did NaN get printed to the console? Keep going to find out why and learn how to fix it!

4.

Did `NaN` get printed to the console? `NaN` stands for "Not a Number" in C#. So what happened?

The built-in method `Math.Sqrt()` can only take a positive number as a value, but the value of `numberTwo` is negative. But we can fix it. The method `Math.Abs()` will find the absolute value of a number. A good reminder that when we use built-in methods, to check the documentation so we know how to use them!

Inside of the `Math.Sqrt()` method, add the `Math.Abs()` method, so it takes the variable `numberTwo` as a value. Re-run the code and see what gets printed to the console this time!

Hint

Make sure when you are chaining methods that you have the correct number of parentheses.

```
Math.Floor(Math.Sqrt(Math.Abs(number)));
```