Call a Method

You've been using methods since you started learning C#! Commands like `Console.WriteLine()` and `Math.Min()` are methods.

Each method has a different behavior: The first method prints something to the console, and the second finds the smallest of two given numbers. We activate a method's behavior by *calling* it. In C# we do this by adding parentheses to the end of a method name.

```
Console.WriteLine();
```

Some methods accept inputs called *arguments*. `Console.WriteLine()` accepts one string argument. That argument will be printed to the console.

```
// This prints "I'm hungry!"
Console.WriteLine("I'm hungry!");
```

Other methods accept multiple arguments, like `Math.Min()`. It expects two number inputs.

```
Math.Min(3, 5);
```

You've probably seen built-in methods for each data type too. Every string has access to methods like `IndexOf()` and `Substring()`.

```
string name = "beatrice";
name.Substring(0, 3); // returns "bea"
```

☑**Instructions**

1.

Call `Math.Min()` with two arguments (any two integers will do). You can view the documentation if you're not sure how to use this method.

Hint

You can call a method with two arguments like this:

```
YourMethodName(100, 99);
```

2.
Call `Console.WriteLine()` using the variable `msg` as an argument.

Hint

You can call a method with one argument like this:

```
YourMethodName("hullaballoo");
```

3.
Get the first letter of the `msg` string using `Substring()`. You can view the documentation if you're not sure how to use this method.

Hint

The first argument is where you want to your substring to start. The second argument is the length of the substring.

In this example, we are getting the substring in `"beatrice"` that starts at 0 (the beginning of the string) and is 3 characters long.

```
string name = "beatrice";
name.Substring(0, 3); // returns "bea"
```