Capture Output

Like a math function or a factory machine, a method takes input and *returns* output. We've just seen how input works (arguments). Let's see how output works.

When a method *returns*a value, it essentially passes a piece of data to wherever it was called. One way to capture the returned value of a method is with a variable:

```
int smallerNumber = Math.Min(3, 4);
```

`Math.Min()` returns the value `3`, so you can imagine that value taking its place.

```
int smallerNumber = 3;
```

We can then use that variable as input to other methods, like `Console.WriteLine()`:

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

In this case, we can take a shortcut by nesting the method calls:

```
Console.WriteLine(Math.Min(3, 4));
```

Now the value returned by `Math.Min()` is used as input to `Console.WriteLine()`.

Not every method returns a value. `Console.WriteLine()`, for example, prints `3` to the console but it doesn't pass the value `3` to its caller. If you're not sure what a method returns you can always check the Microsoft documentation.

☑Instructions

1.

The designer of C# is `"Anders Hejlsberg"`. His first name is nice, but we want to print the second name alone.

First, find the index of the space (`" "`) in the string `designer` and store it in a variable `indexOfSpace`.

You'll need to use the `IndexOf()` method.

Hint

In this example, we are looking for the position of the character `e` in the string. The result is `1` because the first position is 0.

```
string name = "beatrice";
int location = name.IndexOf("e"); // sets location variable to 1
```

2.

Use `Substring()` and the variable `indexOfSpace` to get the second name. Store the returned value in a variable `secondName`.

Hint

The first argument is where you want 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";
string shortened = name.Substring(0, 3); // sets variable to "bea"
```

3.
Print `secondName` to the console.

Hint

You can call a method with one argument like this:

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