Main()

Now that you're familiar with classes, you're ready to tackle the `Main()` method, the entry point for any program. You've seen it many times, but now you can explain every part!

```
class Program
{
  public static void Main (string[] args)
  {
  }
}
```

- `Main()` is a method of the `Program` class.

- `public` — The method can be called outside the `Program` class.

- `static` — The method is called from the class name: `Program.Main()`.

- `void` — The method means returns nothing.

- `string[] args` — The method has one parameter named `args`, which is an array of strings.

`Main()` is like any other method you've encountered. It has a special use for C#, but that doesn't mean you can't treat it like a plain old method!

☑Instructions

1.
Each time we run `dotnet run`, the `Main()` method is called. We can include arguments on the command line, like `dotnet run arg1 arg2 arg3` that will be converted into an array as the `args` parameter. In the console, enter:

```
dotnet run mango pineapple lychee
```

Based on this new information, how is your text printed to the console?

| Hint   ⌄ |

When you run the code, `Main()` is called with one argument: an array containing `"mango"`, `"pineapple"`, and `"lychee"`.

This is equivalent to calling the method like this:

```
Program.Main(new string[] {"mango", "pineapple", "lychee"});
```

In the body of `Main()`, the values in the array are converted to a single string and printed to the console.