Static Classes

We covered a few static members: field, property, method, and constructor. What if we made the whole class static?

```
static class Forest {}
```

A static class cannot be instantiated, so you only want to do this if you are making a utility or library, like `Math` or `Console`.

These two common classes are static because they are just tools — they don't need specific instances and they don't store new information.

Now when you see something like:

```
Math.Min(34, 54);
Console.WriteLine("yeehaw!");
```

You know that these are two static classes calling two static methods.

☑**Instructions**

1.

We rarely create static classes of our own, so let's practice using other static classes. First print the value of pi — a commonly-used value in geometry —, which is stored in `Math.PI`.

Hint

Use `Console.WriteLine()` to print the value `Math.PI` to the console.

2.

Find the absolute value of `-32` using the method `Math.Abs()`. This method returns the absolute value, or "positive version", of the argument.

Print the result to the console.