

Expression Trees in C#

Understanding Where Expressions
Come From



Chris B. Behrens

@chrisbbehrens

Version Check



Applicable

- from .NET 6 - 8
- from .C# 10 - 12

Visual Studio 2022

LINQPad



Github Repo

[https://github.com/FeynmanFan/
csharp-expression-trees](https://github.com/FeynmanFan/csharp-expression-trees)



Version Check



This course is 100% applicable to:

- Visual Studio 2022 version 17.1.1,
Community Edition
- LINQPad 7, Free Edition

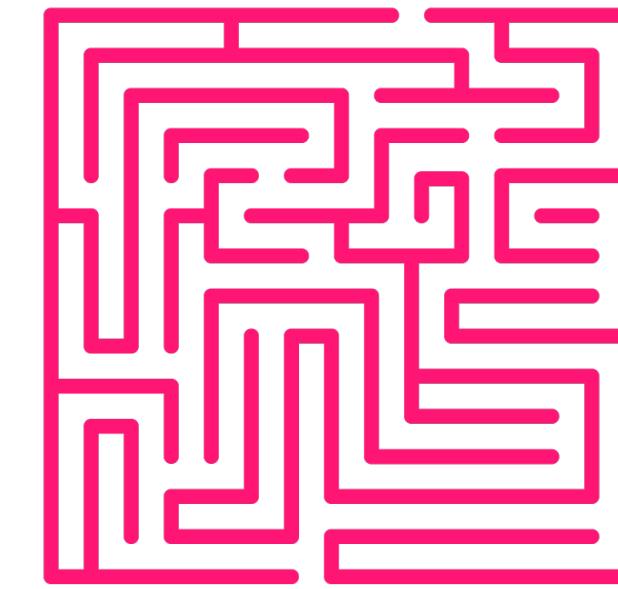
**[https://github.com/FeynmanFan/
csharp-expression-trees](https://github.com/FeynmanFan/csharp-expression-trees)**



This Course Is Unique



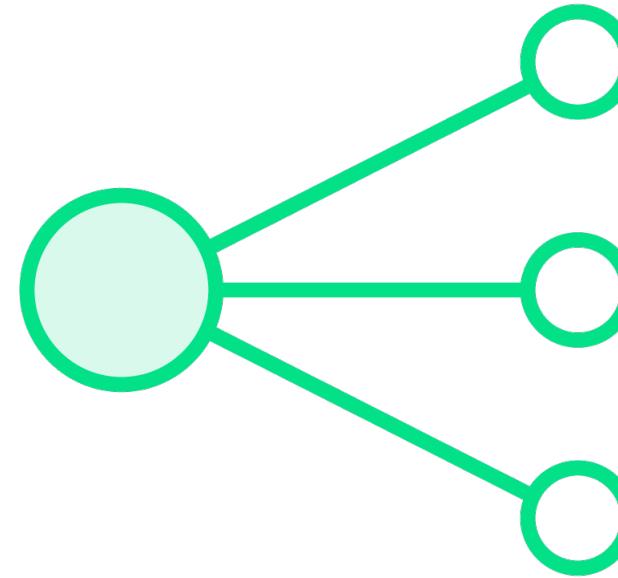
Course #45



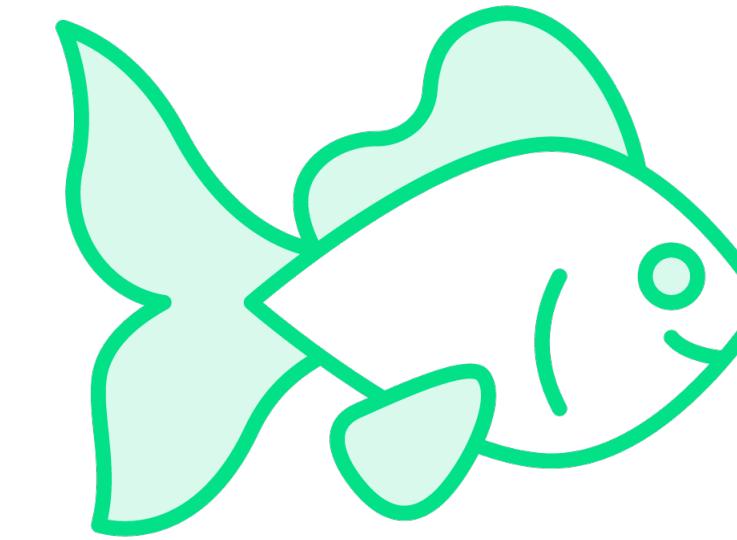
This subject matter
is hard to explain



Fish Don't Know That They're Wet



**Expression trees were the
very first thing you learned**



**But a fish has no
concept of “water”**



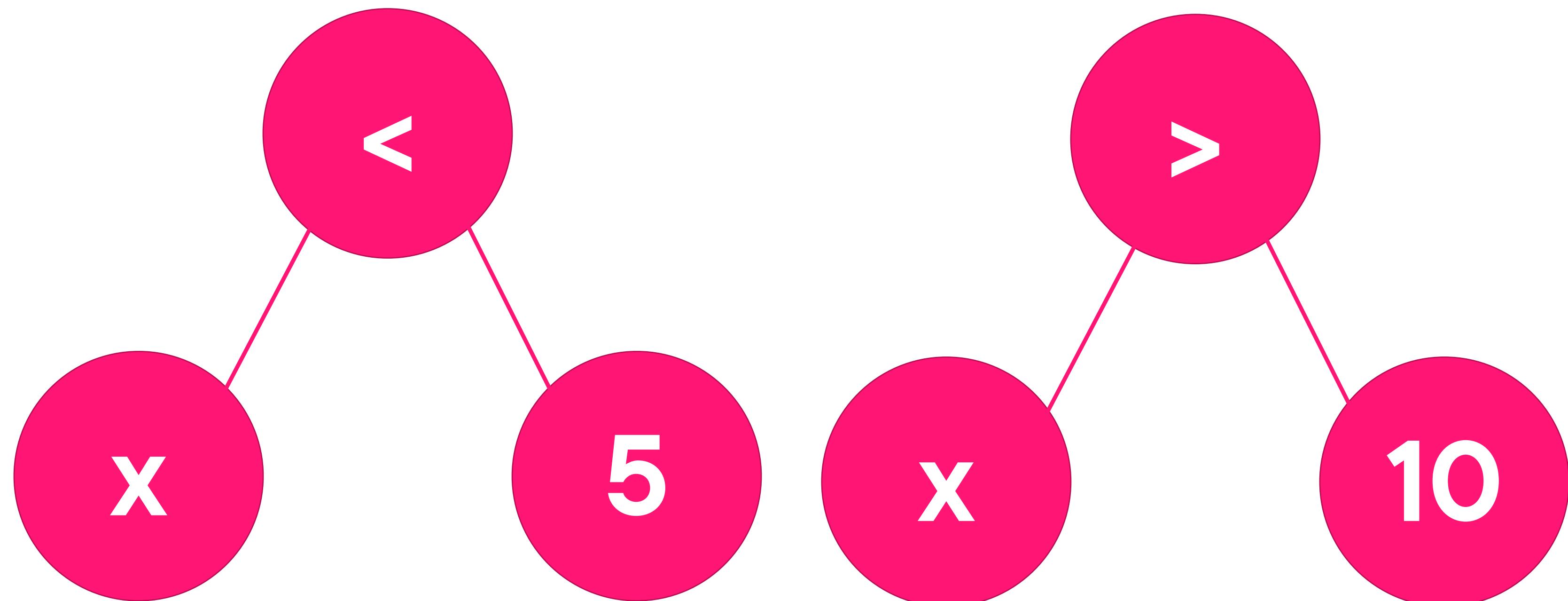
What Is an Expression? Why Is It in a Tree?

```
public bool IsEven(int value){  
    return value % 2 == 0;  
}
```

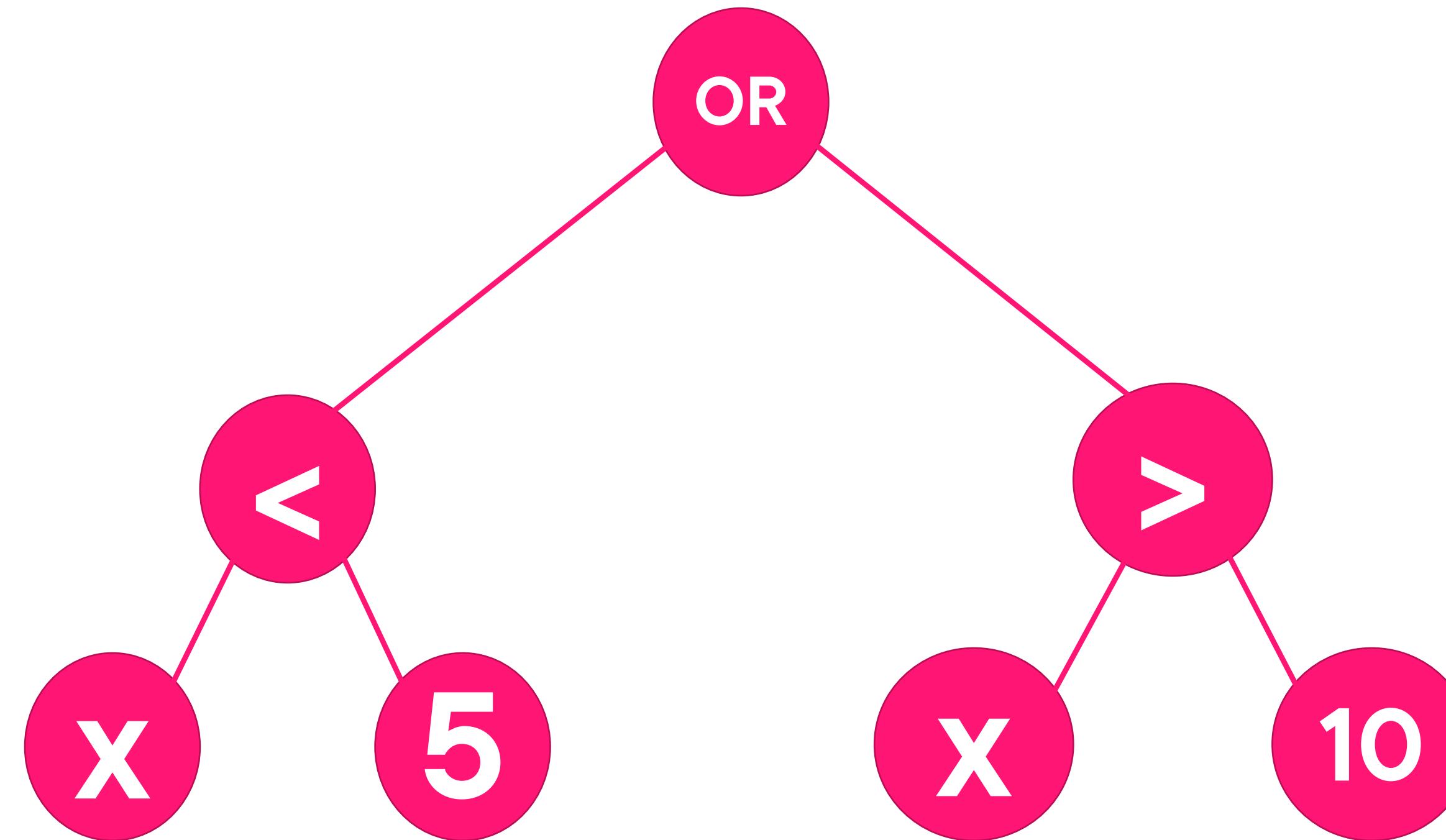
x < 5



Expression Tree for $x < 5$



Expression Tree for $x < 5$



That's All for Now

“A simple binary tree”

Take a second and turn that idea over in your head

Yes, this is tricky to grasp...

We're headed to a fun place



Delegates, Functions, and Expressions



Some Context

You can
understand those
ideas

But not
understand how
they apply in a
given context

So, let's take
another look



Delegates

A delegate is a type that safely encapsulates a method, similar to a function pointer in C and C++.

<https://bit.ly/3CXWOx5>



Simple Delegate

```
public delegate void Del(string message);

// Create a method for a delegate.
public static void DelegateMethod(string message)
{
    Console.WriteLine(message);
}

// Instantiate the delegate.
Del handler = DelegateMethod;

// Call the delegate.
handler("Hello World");
```





What is it that this feels like?

It feels like an interface

We could pass `IMessageWriter` instead

This is inversion of control

**Delegates are a simpler approach, assuming
that you've paid the cognitive price of
understanding it**

Delegates allow you to time shift

That's our ultimate destination



Demo



**Create and observe the delegate example
from the docs**

**Create a slightly more complicated
example**

Talk about what it means



We're Pushing Design-time to Run-time

We're “writing” code
at run-time

Our run-time is the
user's design-time

We've deferred the location

Again, like an interface



The Func and Action Delegates in C#



Our Code

```
public delegate void Del(string message);

// Create a method for a delegate.
public static void DelegateMethod(string message)
{
    Console.WriteLine(message);
}

// Instantiate the delegate.
Del handler = DelegateMethod;

// Call the delegate.
handler("Hello World");
```



Action Notation

Action action;

Func<string> func;

Action<string> action();

Func<void> func



How Generics Fit into This

Hat tip to
John H. Müller

Func<void> and
Func<> are out

Because they
would require non-
standard generic
notation



Demo



- Refactor our delegate to an Action
- Modify our function to return a result
- Write a function that takes that function as an argument
- Put it all together
- Watch it work



“Generic” Is the Magic Word

Both in the plain language sense

And in a technical sense

We'll rely on this stuff as we move forward



Lambda Functions



What Is Lambda?

$$f(x) = x^2$$

$$x \mapsto x^2$$

```
private double Square(double value)
{
    return Math.Pow(value, 2);
}
```

x => Math.Pow(x, 2);



Lambda Representations

We don't need
name, scope, or
parameter names

Because it lacks
these elements...

We can call it an
*anonymous
function*

Lambdas and
anonymous
functions are not
exactly the same...

But for our
purposes, they are



Demo



Rewrite our simple app for a third time

Using lambdas

Talk about what it means



One More Step

We've detached a pure representation of the logic

We'll close the gap as we move ahead



Summary



Expression Trees

What expressions are

Why we put them in trees

Delegates

Actions and Functions

Lambdas

A simple console app



The Other Value Proposition

**Understanding the internals
can be critical**

**Otherwise, you are
subject to decisions
the platform-makers made**

