INSIDE IQUERYABLE

THE POWER ⚡ OF .NET EXPRESSIONS

@JEREMYLIKNESS

@CLEVERSNTGEEK

@CLEVERSNTGEEK

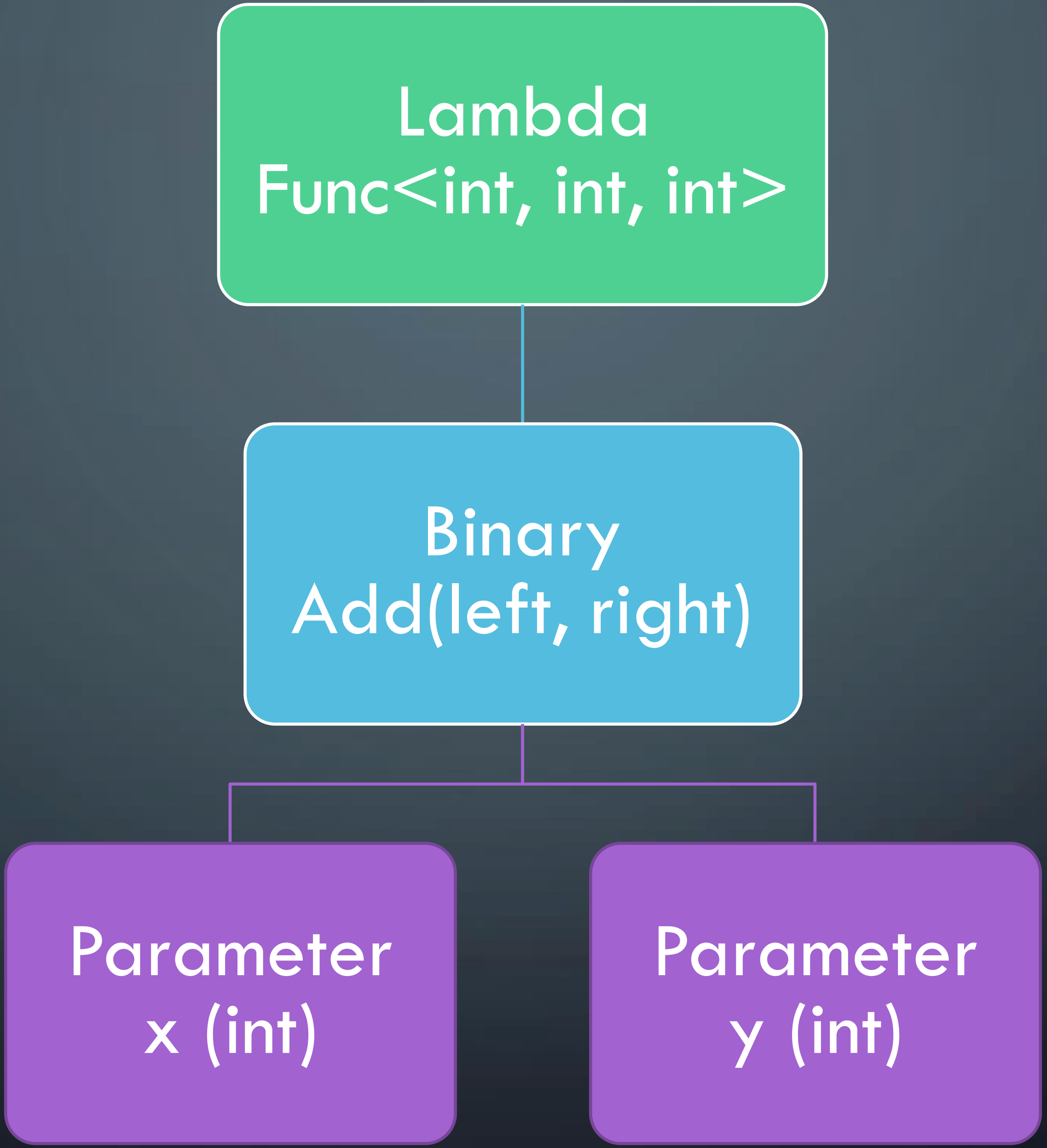
WHY EXPRESSIONS?

DEMO: BLAZOR WEBASSEMBLY

40 + 2

DEMO: INTRO TO EXPRESSIONS

01-INTRO-EXPRESSIONS.IPYNB



Lambda
Func<int, int, int>

Binary
Add(left, right)

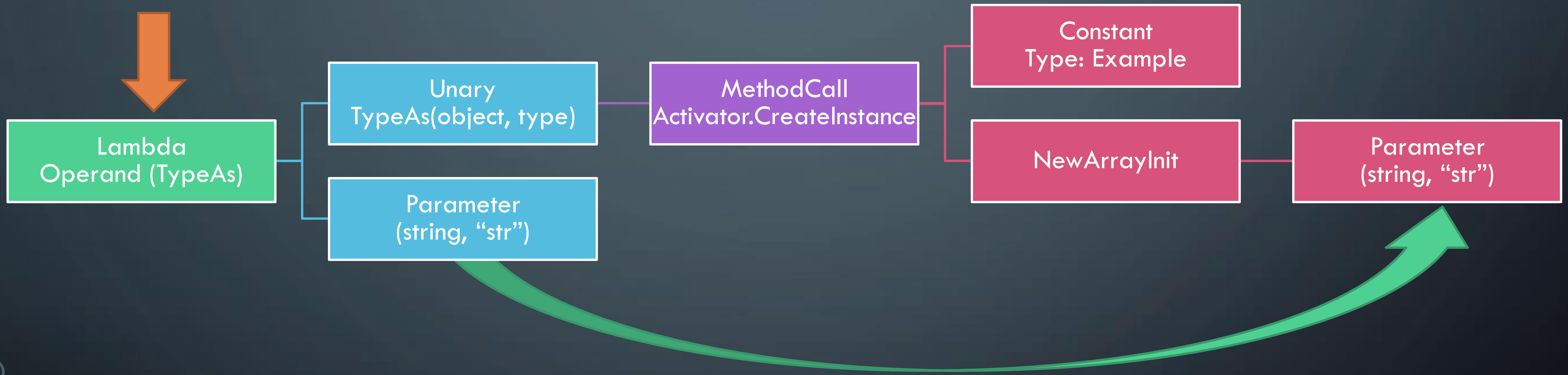
Parameter
x (int)

Parameter
y (int)

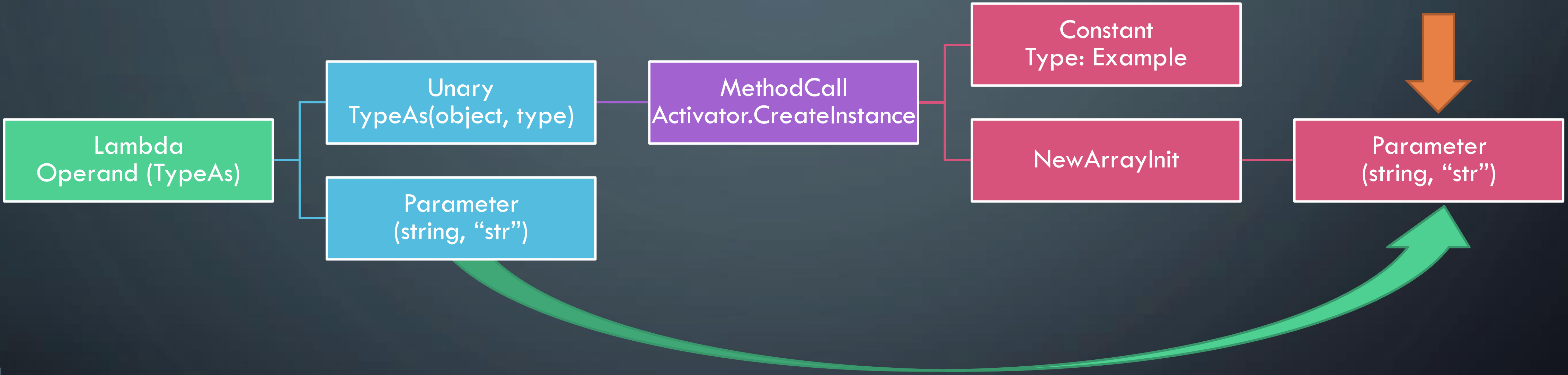
DEMO: CONSTRUCTORS

02-CONSTRUCTOR-EXAMPLE.IPYNB

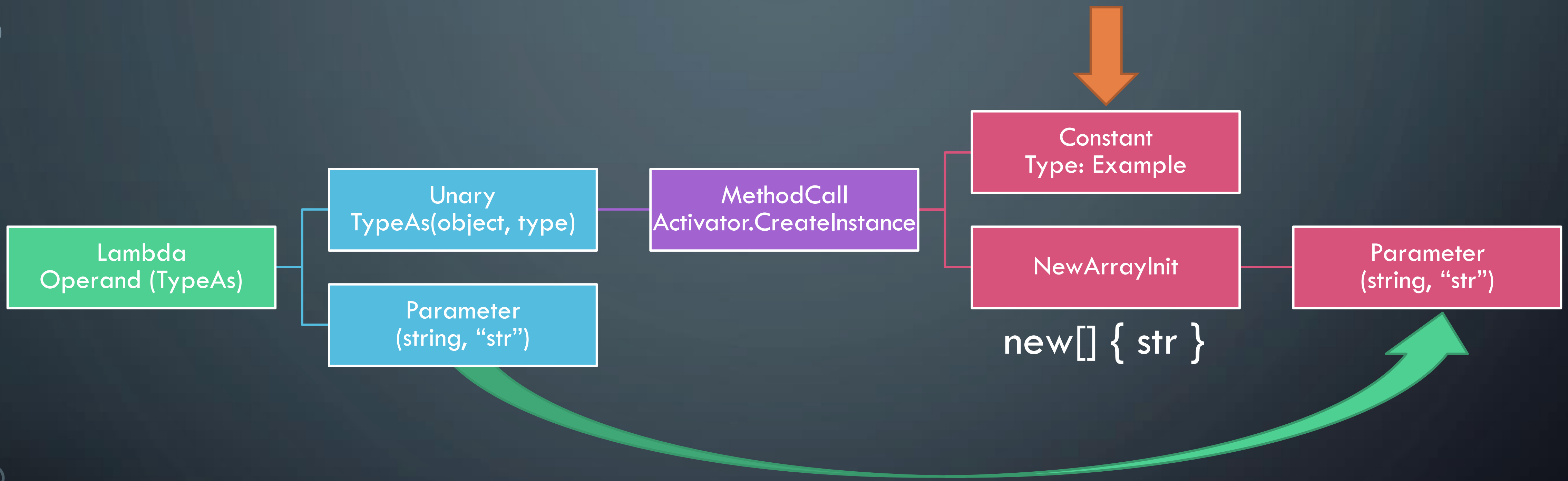
```
Expression<Func<string, Example>> expr4 =
str => Activator.CreateInstance(typeof(Example), str) as Example;
```



```
Expression<Func<string, Example>> expr4 =
str => Activator.CreateInstance(typeof(Example), str) as Example;
```



```
Expression<Func<string, Example>> expr4 =
str => Activator.CreateInstance(typeof(Example), str) as Example;
```



Lambda
Operand (TypeAs)

Unary
TypeAs(object, type)

Parameter
(string, “str”)

MethodCall
Activator.CreateInstance

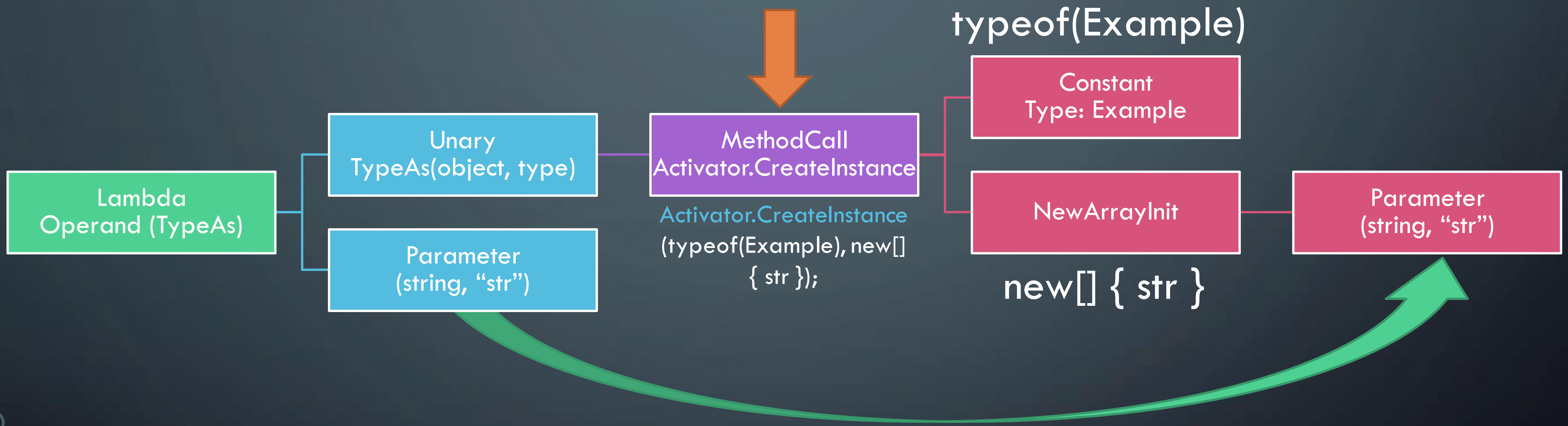
Constant
Type: Example

NewArrayInit

Parameter
(string, “str”)

new[] { str }

```
Expression<Func<string, Example>> expr4 =
str => Activator.CreateInstance(typeof(Example), str) as Example;
```



typeof(Example)

Constant
Type: Example

Lambda
Operand (TypeAs)

Unary
TypeAs(object, type)

Parameter
(string, “str”)

MethodCall
Activator.CreateInstance

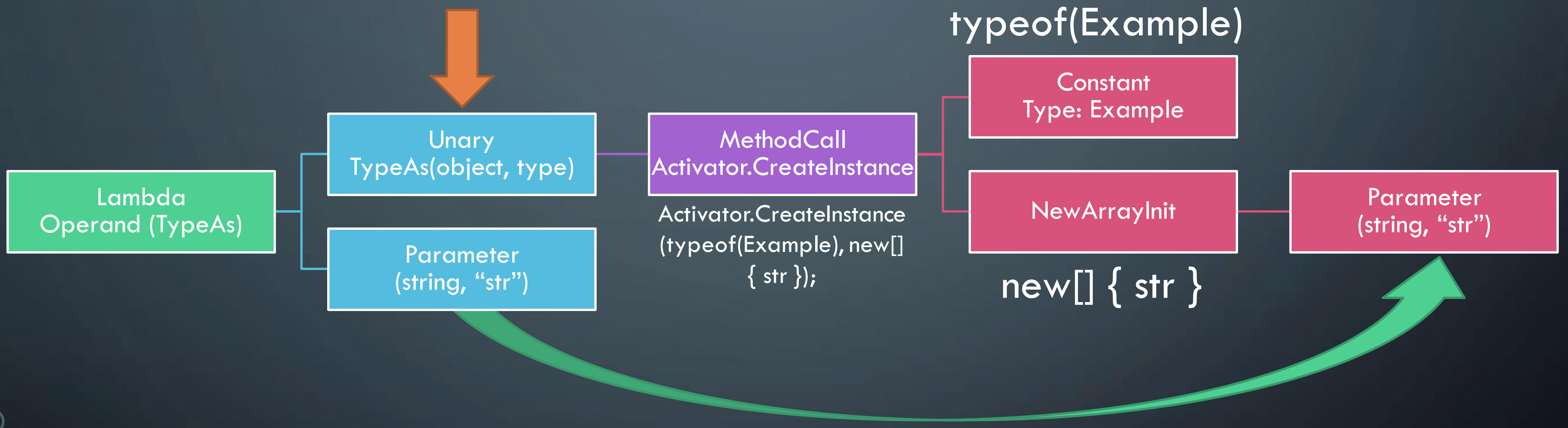
Activator.CreateInstance
(typeof(Example), new[]
{ str });

NewArrayInit

Parameter
(string, “str”)

new[] { str }

```
Expression<Func<string, Example>> expr4 =
str => Activator.CreateInstance(typeof(Example), str) as Example;
```



typeof(Example)

Lambda
Operand (TypeAs)

Unary
TypeAs(object, type)

Parameter
(string, “str”)

MethodCall
Activator.CreateInstance

Activator.CreateInstance
(typeof(Example), new[]
{ str });

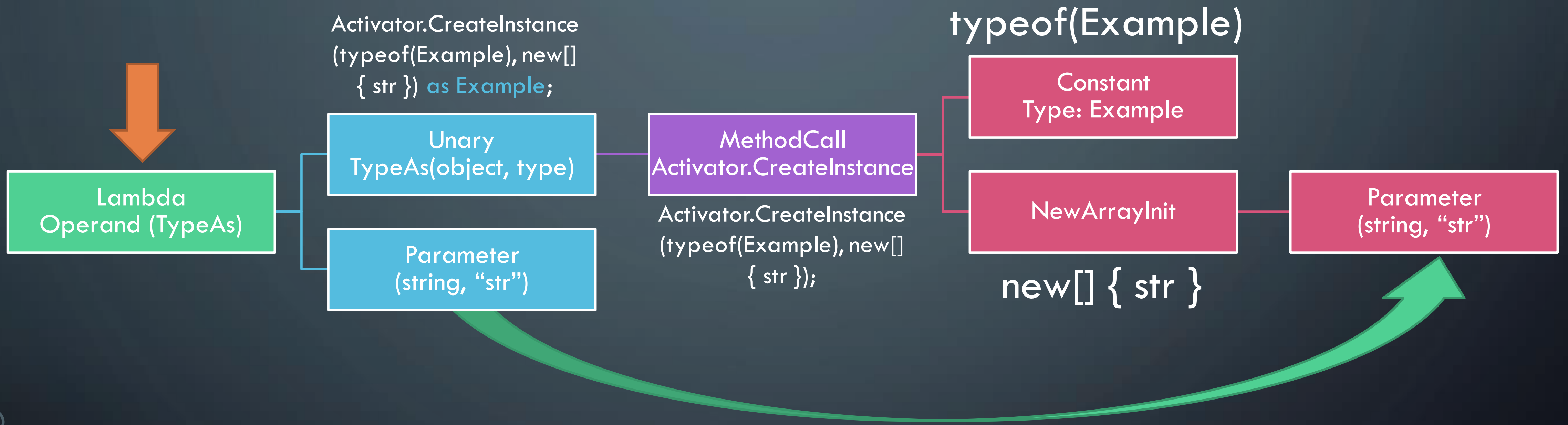
Constant
Type: Example

NewArrayInit

new[] { str }

Parameter
(string, “str”)

```
Expression<Func<string, Example>> expr4 =
str => Activator.CreateInstance(typeof(Example), str) as Example;
```



Activator.CreateInstance (typeof(Example), new[] { str }) as Example;

Lambda
Operand (TypeAs)

Unary
TypeAs(object, type)

Parameter
(string, “str”)

MethodCall
Activator.CreateInstance

Activator.CreateInstance (typeof(Example), new[] { str });

typeof(Example)

Constant
Type: Example

NewArrayInit

new[] { str }

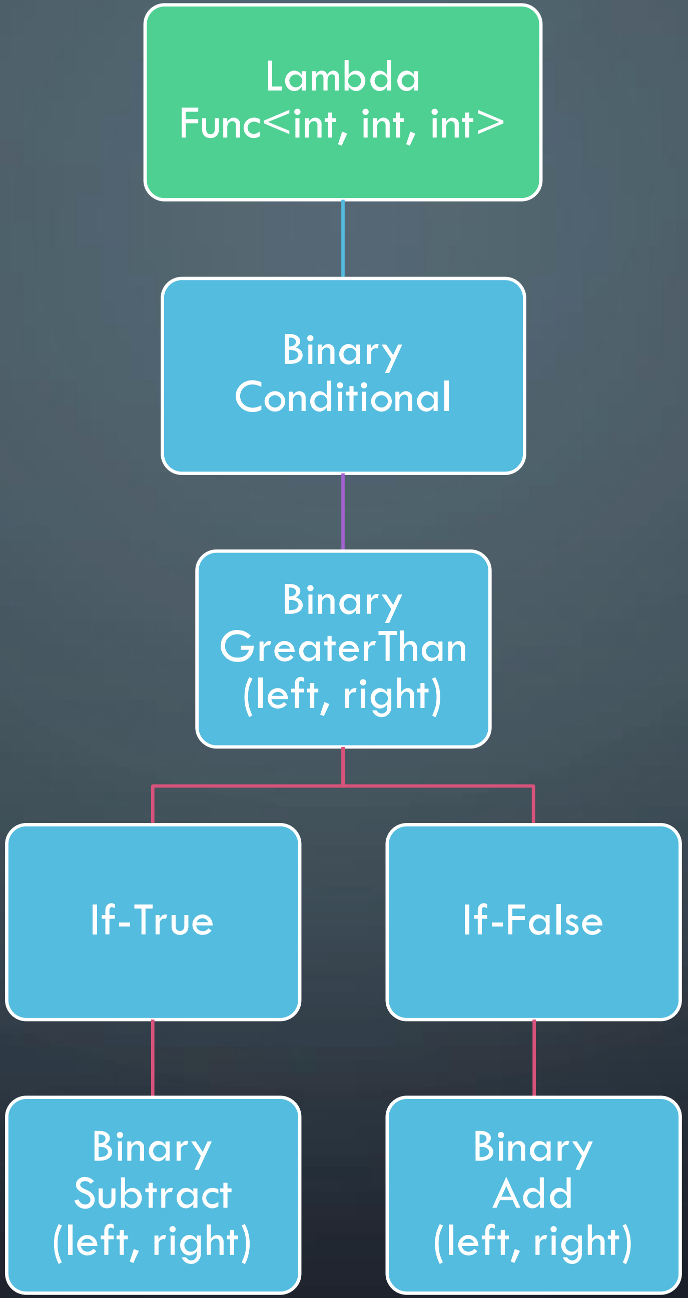
Parameter
(string, “str”)

BUT WHAT ABOUT PERFORMANCE?

BENCHMARKING NEW() VS. ACTIVATOR.CREATEINSTANCE() VS. EXPRESSION.NEW()

ADVANCED EXPRESSIONS

03-ADVANCED-EXPRESSION.IPYNB



Lambda
Func<int, int, int>
Binary
Conditional
Binary
GreaterThan
(left, right)
If-True
If-False
Binary
Subtract
(left, right)
Binary
Add
(left, right)

EXPRESSION RECAP

01

Expressions are building blocks for declarative code

EXPRESSION RECAP

01

Expressions are building blocks for declarative code

02

Expressions perform at speeds close to native code

EXPRESSION RECAP

01

Expressions are building blocks for declarative code

02

Expressions perform at speeds close to native code

03

Expression trees are immutable and represent the *definition* of code to run

EXPRESSION RECAP

01

Expressions are building blocks for declarative code

02

Expressions perform at speeds close to native code

03

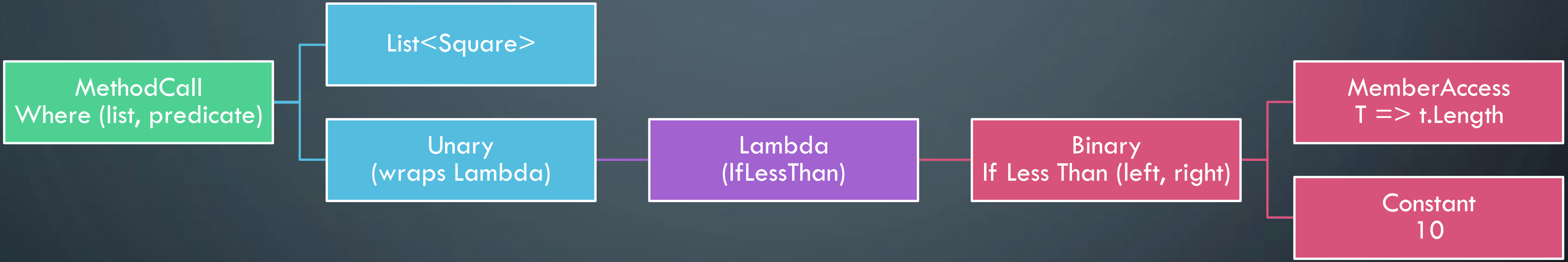
Expression trees are immutable and represent the *definition* of code to run

04

Lambda expressions help turn expression trees into the *implementation* of code to run via "Compile"

INTRODUCING IQUERYABLE

04-INTRODUCING-IQUERYABLE.IPYNB



MethodCall
Where (list, predicate)

List<Square>

Unary
(wraps Lambda)

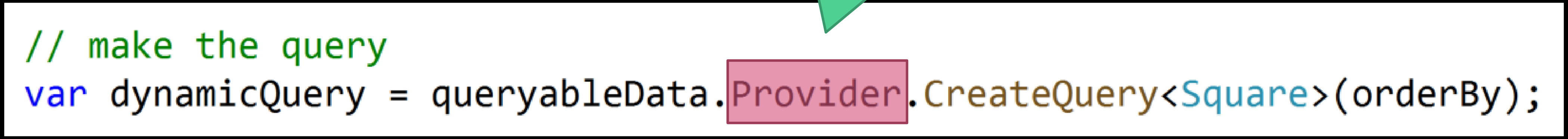
Lambda
(IfLessThan)

Binary
If Less Than (left, right)

MemberAccess
T => t.Length

Constant
10

Here Be Dragons



```
// make the query
var dynamicQuery = queryableData.Provider.CreateQuery<Square>(orderBy);
```

WHY IQUERYABLE?

The provider is magic

Add metadata and state

Dynamic expressions

Snapshots before/after the tree is implemented

Inspect the tree and do something different… like generate SQL (EF Core)

Mutate the tree (wait, isn't it immutable?)

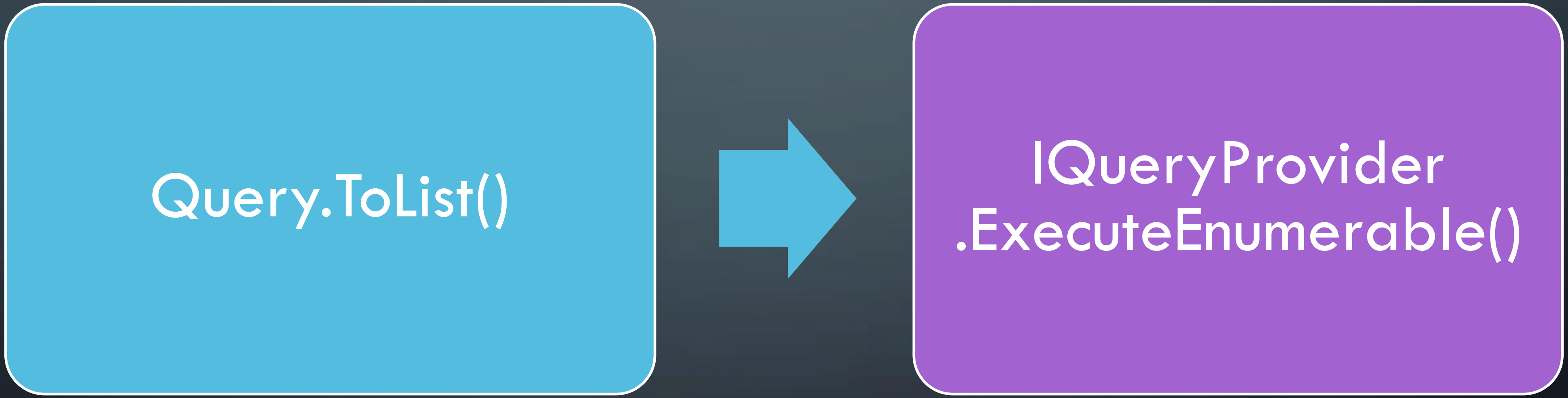
PROVIDER AND SNAPSHOT

05-PROVIDER-AND-SNAPSHOTS.IPYNB

IQUERYPROVIDER

Query.ToList()

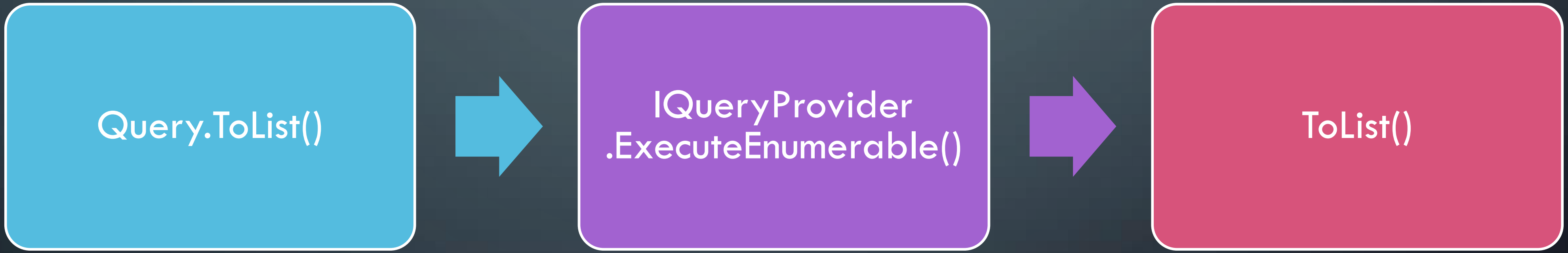
IQUERYPROVIDER



Query.ToList()

IQueryProvider
.ExecuteEnumerable()

IQUERYPROVIDER



Query.ToList()

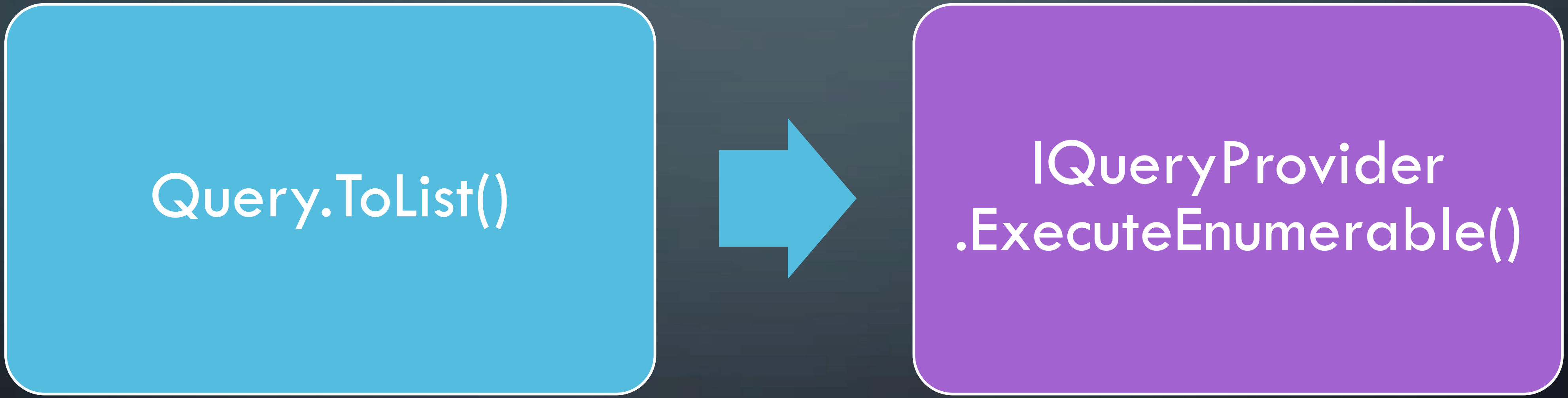
IQueryProvider
.ExecuteEnumerable()

ToList()

ISNAPSHOTQUERYPROVIDER

Query.ToList()

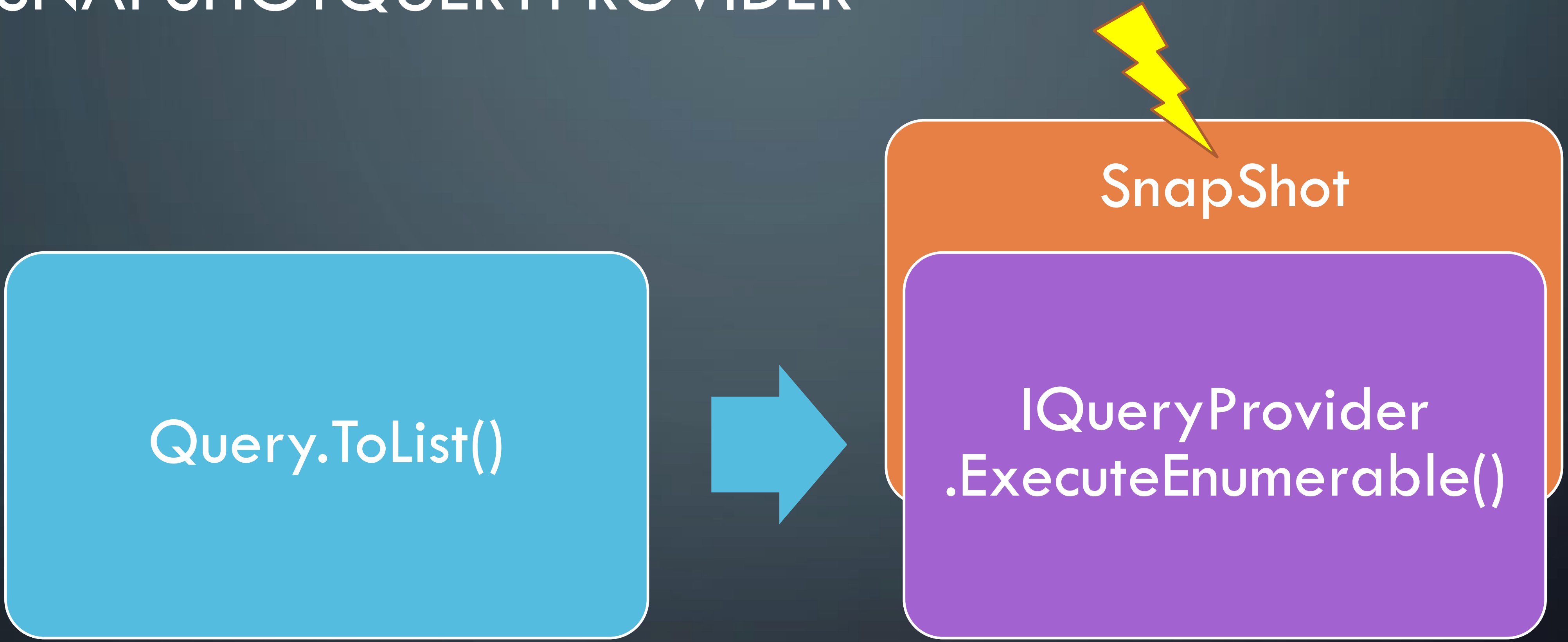
ISNAPSHOTQUERYPROVIDER



Query.ToList()

IQueryProvider
.ExecuteEnumerable()

ISNAPSHOTQUERYPROVIDER

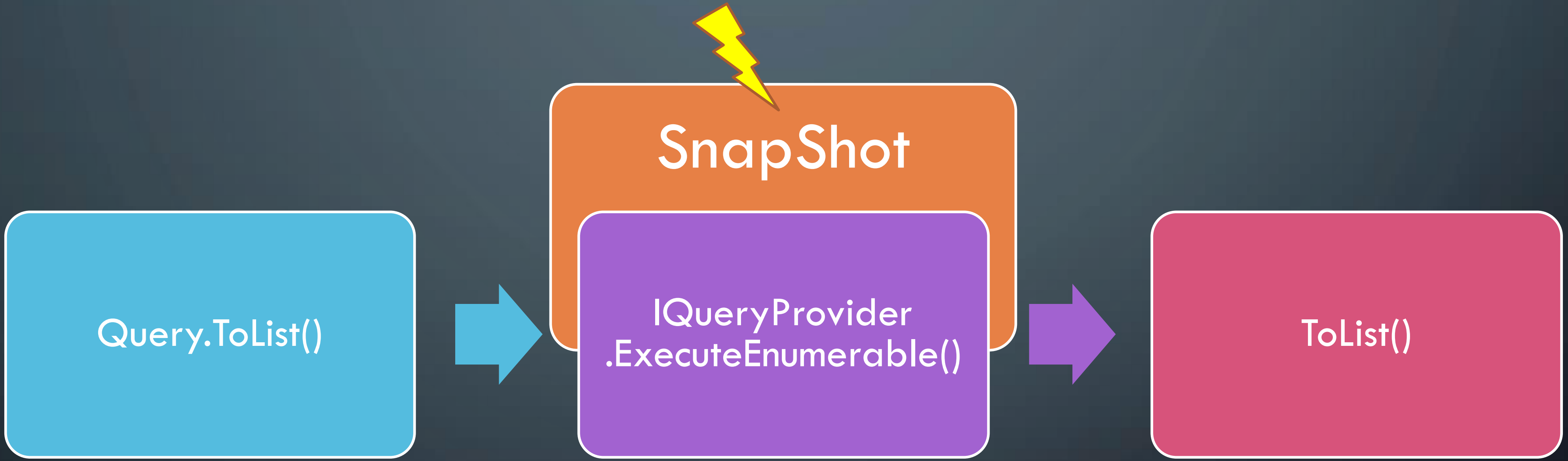


Query.ToList()

SnapShot

IQueryProvider
.ExecuteEnumerable()

ISNAPSHOTQUERYPROVIDER



https://github.com/JeremyLikness/ExpressionPowerTools/blob/master/src/ExpressionPowerTools.Core/Providers/QuerySnapshotProvider.cs

MUTATION

THE QUERY TRANSFORMER (QUERYMUTATORS)

THANK YOU!

@JEREMYLIKNESS

HTTPS://GITHUB.COM/JEREMYLIKNESS/IQUERYABLEEXPRESSIONEXAMPLES

HTTPS://GITHUB.COM/JEREMYLIKNESS/EXPRESSIONPOWERTOOLS