

# Learn to Love Lambdas

## And LINQ, Too!

Jeremy Clark  
[www.jeremybytes.com](http://www.jeremybytes.com)  
[@jeremybytes](https://twitter.com/jeremybytes)

# COMMON USES

- LINQ (Language Integrated Query)

```
if (NameFilterCheckBox.IsChecked.Value)
    people = people.Where(p => p.FirstName == NameTextBox.Text);
```

- Func<T> and Action<T>

```
Func<Person, string> personFormatter = p => p.LastName.ToUpper();
```

- Callbacks and Event Handlers

```
var proxy = new PersonServiceClient();
proxy.GetPeopleCompleted +=
    (s, a) => PersonListBox.ItemsSource = a.Result;
proxy.GetPeopleAsync();
```

# ANATOMY

**3 Parts**    Parameter(s)  
              => “goes to” operator  
              Expression or Statements

## Expression Lambda

```
(Person person) => person.FirstName == "John"
```

## Statement Lambda

```
(object sender, EventArgs args) =>  
{  
    ListBox.ItemsSource = args.Result;  
    Console.WriteLine("Hello");  
}
```

# ANATOMY

**3 Parts** Parameter(s)  
=> “goes to” operator  
Expression or Statements

## Expression Lambda

```
(Person p) => p.FirstName == "John"
```

## Statement Lambda

```
(object s, EventArgs a) =>  
{  
    ListBox.ItemsSource = a.Result;  
    Console.WriteLine("Hello");  
}
```

**Single Character  
Parameter Names**

# SYNTACTIC VARIATIONS

- Optional Parameter Types (Parameter Type Inference)  
 $(s, e) \Rightarrow \{ \text{ListBox.ItemsSource} = e.\text{Result}; \}$
- Optional Parentheses for a Single Parameter  
 $e \Rightarrow \{ \text{ListBox.ItemsSource} = e.\text{Result}; \}$
- Optional Braces for a Single Statement  
 $e \Rightarrow \text{ListBox.ItemsSource} = e.\text{Result};$
- Empty Parentheses for No Parameter  
 $() \Rightarrow \{ \text{Console.Write("John");} \}$

# QUERY SYNTAX VS FLUENT SYNTAX

## Query Syntax

```
from p in people  
where p.FirstName == NameTextBox.Text  
orderby p.LastName  
select p;
```

## Fluent Syntax

```
people.Where(p => p.FirstName == NameTextBox.Text)  
    .OrderBy(p => p.LastName);
```



# LINQ METHODS

```
public static IEnumerable<TSource> Where<TSource>(
    this IEnumerable<TSource> source,
    Func<TSource, bool> predicate
)
```

Put Your Lambda Expression Here



# RESOURCES

## Code Samples & Resources

[https://github.com/jeremybytes/  
sdd-2022](https://github.com/jeremybytes/sdd-2022)





THANK YOU!

Jeremy Clark

- <http://www.jeremybytes.com>
- [jeremy@jeremybytes.com](mailto:jeremy@jeremybytes.com)
- @jeremybytes