



LINQ

It's not just for databases

Jeremy Clark

jeremybytes.com

Language Integrated Query (LINQ)

- What it is
- Query syntax vs. Method syntax
- Understanding parameters
- A few methods
 - Where
 - OrderBy, ThenBy
 - Average
 - Min, Max, MinBy, MaxBy
 - First, FirstOrDefault, Single, SingleOrDefault

What is LINQ?

- Query in C#
- Filter, Sort, Aggregate
- Works with IQueryable<T>
 - Databases (Entity Framework / SQL)
- Works with IEnumerable<T>
 - Objects
- Lazy Evaluation

Not Just For Databases

Query Syntax vs. Method Syntax

- Query Syntax

```
from p in people
where p.GivenName == NameFilterValue
orderby p.FamilyName
select p;
```

- Method Syntax

```
people.Where(p => p.GivenName == NameFilterValue)
        .OrderBy(p => p.FamilyName);
```

Query Syntax vs. Method Syntax

- Query Syntax

- Supports a limited number of functions (11)

<https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/standard-query-operators-overview#query-expression-syntax-table>

- Method Syntax

- Supports all of the functions

<https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic.ienumerable-1?view=net-8.0#extension-methods>

Anatomy of a LINQ Method

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

Takes IEnumerable<TSource>
Returns IEnumerable<TSource>

Anatomy of a LINQ Method

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

Extension method
Will work on any IEnumerable<T>

Anatomy of a LINQ Method

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

Delegate Parameter

```
public delegate TResult Func<in T, out TResult>(T arg);
```


Generics Filled In

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

```
public delegate bool Func<Person, bool>(Person arg);
```

Comparer Methods

```
OrderBy<...>(..., Func<TSource, TKey> keySelector);
```

Implements IComparable<T>

Comparer Methods

```
OrderBy<...>(..., Func<TSource, TKey> keySelector);
```

```
OrderBy<...>(..., Func<TSource, TKey> keySelector,  
             IComparer<TKey> comparer);
```

```
IComparer<T>  
    public int Compare(T? x, T? y);
```

Comparer Methods

```
OrderBy<...>(..., Func<TSource, TKey> keySelector);
```

```
ThenBy<...>(..., Func<TSource, TKey> keySelector);
```

```
Min<...>(..., Func<TSource, TResult> selector);
```

```
MaxBy<...>(..., Func<TSource, TResult> selector);
```

Equatable Methods

```
GroupBy<...>(..., Func<TSource, TKey> keySelector);
```

```
Join<...>(..., Func<TOuter, TKey> outer,  
          Func<TInner, TKey> inner);
```

Implements IEquatable<T>



Thank You!

Jeremy Clark

- jeremybytes.com
- youtube.com/jeremybytes

<https://github.com/jeremybytes/linq-in-memory>