# LINQ



**Deborah Kurata**CONSULTANT | SPEAKER | AUTHOR

@deborahkurata | blogs.msmvps.com/deborahk/

# LINQ

Language INtegrated Query

A way to express queries against a data source directly from a .NET language, such as C#.



#### Data Sources: LINQ Provider





#### Query Syntax

```
var vendorQuery = from v in vendors
                 where v.CompanyName.Contains("Toy")
                  orderby v.CompanyName
                  select v;
                   Method Syntax
var vendorQuery = vendors
              .Where(v => v.CompanyName.Contains("Toy"))
              .OrderBy(v=> v.CompanyName);
```



# Extension Method

A method added to an existing type without modifying the original type.

"Object-Oriented Programming Fundamentals in C#"



# Delegate

A type that represents a reference to a method with a specific parameter list and return type.

Used to pass methods as arguments to other methods.

```
vendors.Where(Func<Vendor, bool> predicate);
vendors.Where(FilterCompanies);
private bool FilterCompanies(Vendor v)
{
   return v.CompanyName.Contains("Toy");
}
```



# Lambda Expression

A method that can be passed as an argument to a method when that argument is expecting a delegate type.

Useful for writing LINQ query expressions.

vendors.Where(v => v.CompanyName.Contains("Toy"))



#### Overview



Building a LINQ Query: Query Syntax
Building a LINQ Query: Method Syntax
Using Lambda Expressions
LINQ and Collections

**FAQ** 



### Building a LINQ Query: Query Syntax



## Building a LINQ Query: Method Syntax



#### Building a LINQ Query: Method Syntax



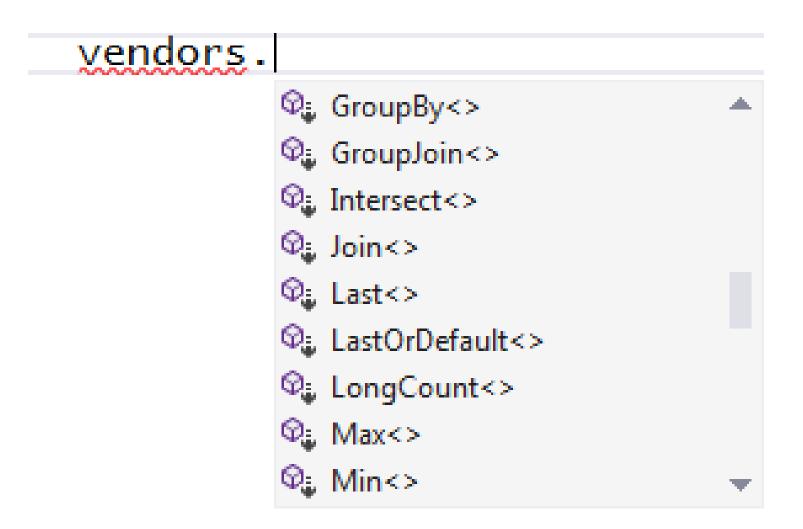
#### Building a LINQ Query: Method Syntax



#### Using Lambda Expressions

```
var vendorQuery = vendors
    .Where(v => v.CompanyName.Contains("Toy"))
    .OrderBy(v => v.CompanyName);
```

#### LINQ and Collections





```
vendors.Where(v =>
         v.CompanyName.Contains("Toy")
vendors.OrderBy(v => v.CompanyName)
vendors.Select(v=>v.Email);
vendors.GroupBy(v => v.Category);
vendors.Average(v => v.NoOfProducts);
vendors.First(v => v.VendorId == 22);
vendors.FirstOrDefault(v =>
                   v.VendorId == 22);
vendors.Where(v =>
        v.CompanyName.Contains("Toy"))
  .GroupBy(v => v.Category)
  .Select(g =>
          g.Sum(v => v.NoOfProducts));
```

- Filter based on criteria
- **◄** Sort on defined string
- **◄** Shape by selecting properties
- Group by a defined property
- Aggregate elements
- Return the first matching element
- Return the first matching element or null
- Combine as needed



#### LINQ with Collections Best Practices

#### Do:

Use LINQ!

Consider using the method syntax over the query syntax

Wait to cast the result to a concrete type until all query operations are defined

#### **Avoid:**

Iterating over the collection more than is required

Using First or Last methods
Use FirstOrDefault or
LastOrDefault instead



## Frequently Asked Questions

- What is LINQ?
  - Language INtegrated Query.
  - A way to express queries against a data source from a .NET language such as C#.
- What type of data source works with LINQ?
  - Any data source that has a LINQ provider.
  - LINQ to Objects, LINQ to SQL, LINQ to Entities (Entity Framework)



#### Frequently Asked Questions (cont)

- What is an extension method?
  - A method added to an existing type without modifying the original type.
  - The LINQ standard query operators are all extension methods.
- What type(s) do the LINQ extension methods extend?
  - IEnumerable and IEnumerable<T>.



#### Frequently Asked Questions (cont)

- What is a delegate?
  - A data type that represents a reference to a method with a specific parameter list and return type.
  - Used to pass methods as arguments to other methods.
- What is a Lambda expression?
  - A method that can be passed as an argument to a method when that argument expects a delegate type.
  - Are helpful for writing LINQ query expressions.



#### Summary



Building a LINQ Query: Query Syntax

**Building a LINQ Query: Method Syntax** 

**Using Lambda Expressions** 

**LINQ and Collections** 



# For more information on LINQ

"Practical LINQ"

"LINQ Fundamentals"

"Querying the Entity Framework"

