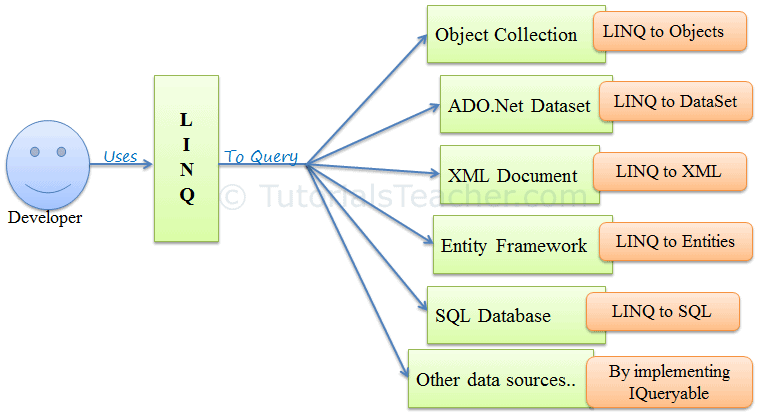
What is LINQ?

LINQ (Language Integrated Query) is uniform query syntax in C# and VB.NET to retrieve data from different sources and formats. It is integrated in C# or VB, thereby eliminating the mismatch between programming languages and databases, as well as providing a single querying interface for different types of data sources.

For example, SQL is a Structured Query Language used to save and retrieve data from a database. In the same way, LINQ is a structured query syntax built in C# and VB.NET to retrieve data from different types of data sources such as collections, ADO.Net DataSet, XML Docs, web service and MS SQL Server and other databases.

[](https://www.tutorialsteacher.com/Content/images/linq/linq-usage.PNG)LINQ Usage

LINQ queries return results as objects. It enables you to uses object-oriented approach on the result set and not to worry about transforming different formats of results into objects.

[](https://www.tutorialsteacher.com/Content/images/linq/linq-execution.PNG)

The following example demonstrates a simple LINQ query that gets all strings from an array which contains 'a'.

Example: LINQ Query to Array

// Data source

string[] names = {"Bill", "Steve", "James", "Mohan" };

// LINQ Query

var myLinqQuery = from name in names

where name.Contains('a')

select name;

// Query execution

foreach(var name in myLinqQuery)

Console.Write(name + " ");

In the above example, string array names is a data source. The following is a LINQ query which is assigned to a variable myLinqQuery.

from name in names

where name.Contains('a')

select name;

You will not get the result of a LINQ query until you execute it. LINQ query can be execute in multiple ways, here we used foreach loop to execute our query stored in myLinqQuery. The foreach loop executes the query on the data source and get the result and then iterates over the result set.

# LINQ Query Syntax

There are two basic ways to write a LINQ query to IEnumerable collection or IQueryable data sources.

1. Query Syntax or Query Expression Syntax
2. Method Syntax or Method Extension Syntax or Fluent

## **Query Syntax**

Query syntax is similar to SQL (Structured Query Language) for the database. It is defined within the C# or VB code.

LINQ Query Syntax:

from *<range variable>* in *<IEnumerable<T> or IQueryable<T> Collection>*

<Standard Query Operators> *<lambda expression>*

<select or groupBy operator> *<result formation>*

The LINQ query syntax starts with from keyword and ends with select keyword. The following is a sample LINQ query that returns a collection of strings which contains a word "Tutorials".

Example: LINQ Query Syntax in C#

// string collection

IList<string> stringList = new List<string>() {

"C# Tutorials",

"VB.NET Tutorials",

"Learn C++",

"MVC Tutorials" ,

"Java"

};

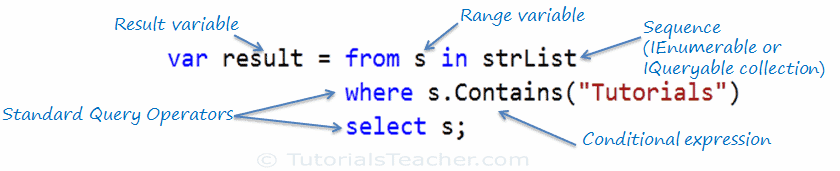
// LINQ Query Syntax

var result = from s in stringList

where s.Contains("Tutorials")

select s;

The following figure shows the structure of LINQ query syntax.

[](https://www.tutorialsteacher.com/Content/images/linq/linq-query-syntax.png)LINQ Query Syntax

Query syntax starts with a From clause followed by a Range variable. The From clause is structured like "From rangeVariableName in IEnumerablecollection".It is similar to a foreach loop: foreach(Student s in studentList).

After the From clause, you can use different Standard Query Operators to filter, group, join elements of the collection. There are around 50 Standard Query Operators available in LINQ. In the above figure, we have used "where" operator (aka clause) followed by a condition. This condition is generally expressed using [lambda expression](https://www.tutorialsteacher.com/linq/linq-lambda-expression).

LINQ query syntax always ends with a Select or Group clause. The Select clause is used to shape the data. You can select the whole object as it is or only some properties of it. In the above example, we selected the each resulted string elements.

In the following example, we use LINQ query syntax to find out teenager students from the Student collection (sequence).

Example: LINQ Query Syntax in C#

// Student collection

class Student

{

public int StudentID,Age;

public string StudentName;

}

IList<Student> studentList = new List<Student>() {

new Student() { StudentID = 1, StudentName = "John", Age = 13} ,

new Student() { StudentID = 2, StudentName = "Moin", Age = 21 } ,

new Student() { StudentID = 3, StudentName = "Bill", Age = 18 } ,

new Student() { StudentID = 4, StudentName = "Ram" , Age = 20} ,

new Student() { StudentID = 5, StudentName = "Ron" , Age = 15 }

};

// LINQ Query Syntax to find out teenager students

var teenAgerStudent = from s in studentList

where s.Age > 12 && s.Age < 20

select s;

# LINQ Method Syntax

Method syntax (also known as fluent syntax) uses extension methods included in the [Enumerable](https://msdn.microsoft.com/en-us/library/system.linq.enumerable(v=vs.110).aspx)) or [Queryable](https://msdn.microsoft.com/en-us/library/system.linq.queryable(v=vs.110).aspx" \t "_blank) static class, similar to how you would call the extension method of any class.

The compiler converts query syntax into method syntax at compile time.

The following is a sample LINQ method syntax query that returns a collection of strings which contains a word "Tutorials".

Example: LINQ Method Syntax in C#

// string collection

IList<string> stringList = new List<string>() {

"C# Tutorials",

"VB.NET Tutorials",

"Learn C++",

"MVC Tutorials" ,

"Java"

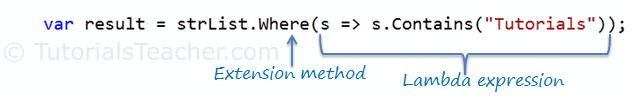
};

// LINQ Query Syntax

var result = stringList.Where(s => s.Contains("Tutorials"));

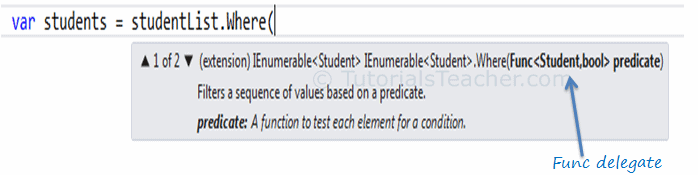
[Try it](https://www.tutorialsteacher.com/codeeditor?cid=cs-5znwGK)

The following figure illustrates the structure of LINQ method syntax.

[](https://www.tutorialsteacher.com/Content/images/linq/linq-method-syntax.png)LINQ Method Syntax Structure

As you can see in the above figure, method syntax comprises of extension methods and Lambda expression. The extension method **Where()** is defined in the Enumerable class.

If you check the signature of the Where extension method, you will find the Where method accepts a [predicate](https://www.tutorialsteacher.com/csharp/csharp-predicate) delegate as Func<Student, bool>. This means you can pass any delegate function that accepts a Student object as an input parameter and returns a Boolean value as shown in the below figure. The lambda expression works as a delegate passed in the Where clause. Learn lambda expression in the next section.

[](https://www.tutorialsteacher.com/Content/images/linq/linq-where-extension-method.png)Func delegate in Where

The following example shows how to use LINQ method syntax query with the IEnumerable<T> collection.

Example: Method Syntax in C#

// Student collection

IList<Student> studentList = new List<Student>() {

new Student() { StudentID = 1, StudentName = "John", Age = 13} ,

new Student() { StudentID = 2, StudentName = "Moin", Age = 21 } ,

new Student() { StudentID = 3, StudentName = "Bill", Age = 18 } ,

new Student() { StudentID = 4, StudentName = "Ram" , Age = 20} ,

new Student() { StudentID = 5, StudentName = "Ron" , Age = 15 }

};

// LINQ Method Syntax to find out teenager students

var teenAgerStudents = studentList.Where(s => s.Age > 12 && s.Age < 20)

.ToList<Student>();

# LINQ Method Syntax

Method syntax (also known as fluent syntax) uses extension methods included in the [Enumerable](https://msdn.microsoft.com/en-us/library/system.linq.enumerable(v=vs.110).aspx)) or [Queryable](https://msdn.microsoft.com/en-us/library/system.linq.queryable(v=vs.110).aspx" \t "_blank) static class, similar to how you would call the extension method of any class.

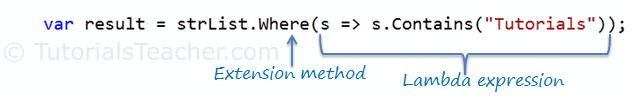
The compiler converts query syntax into method syntax at compile time.

The following is a sample LINQ method syntax query that returns a collection of strings which contains a word "Tutorials".

Example: LINQ Method Syntax in C#

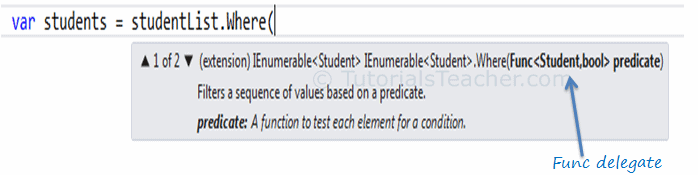
var result = stringList.Where(s => s.Contains("Tutorials"));

The following figure illustrates the structure of LINQ method syntax.

[](https://www.tutorialsteacher.com/Content/images/linq/linq-method-syntax.png)LINQ Method Syntax Structure

As you can see in the above figure, method syntax comprises of extension methods and Lambda expression. The extension method **Where()** is defined in the Enumerable class.

If you check the signature of the Where extension method, you will find the Where method accepts a [predicate](https://www.tutorialsteacher.com/csharp/csharp-predicate) delegate as Func<Student, bool>. This means you can pass any delegate function that accepts a Student object as an input parameter and returns a Boolean value as shown in the below figure. The lambda expression works as a delegate passed in the Where clause. Learn lambda expression in the next section.

[](https://www.tutorialsteacher.com/Content/images/linq/linq-where-extension-method.png)Func delegate in Where

The following example shows how to use LINQ method syntax query with the IEnumerable<T> collection.

Example: Method Syntax in C#

// LINQ Method Syntax to find out teenager students

var teenAgerStudents = studentList.Where(s => s.Age > 12 && s.Age < 20)

.ToList<Student>();