



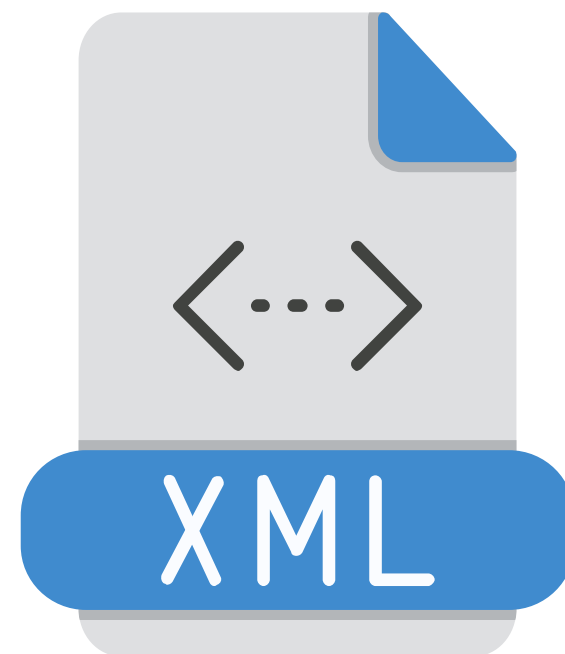
LINQ – Language Integrated Query in .NET

BE-3 CSE SEM 6

MANISH DAMOR - 8023058369 ROLL NO : 04

Introduction to LINQ

- LINQ (Language Integrated Query) is a set of methods in .NET that allows querying data in a declarative way.
- It simplifies querying collections, databases, XML, and other data sources using C#.
- Introduced in .NET Framework 3.5 (2007).

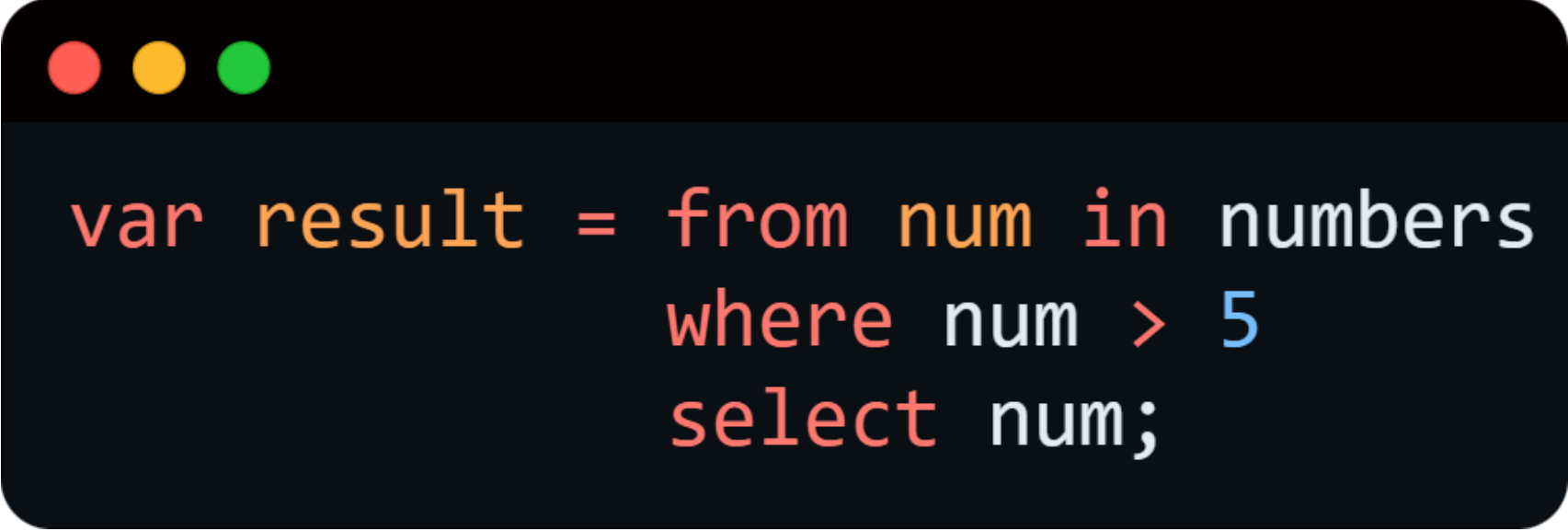


Why Use LINQ?

- Concise & Readable – Reduces boilerplate code
- Type-Safety – Catches errors at compile-time
- Intellisense Support – Works well with Visual Studio
- Works with Different Data Sources – Supports Collections, Databases, XML, JSON
- Improved Performance – Optimized querying

LINQ Syntax & Query Types

Query Syntax (SQL-like)



```
var result = from num in numbers  
              where num > 5  
              select num;
```

Method Syntax (Fluent API, recommended)



```
var result = numbers.Where(num => num > 5).ToList();
```

LINQ vs Traditional Loops

Feature	For Loop	LINQ
Code Length	Longer	Shorter
Readability	Complex	Simple
Performance	Faster for small data	Optimized for large data



```
// Traditional Approach  
List<int> evens = new List<int>();  
foreach (var num in numbers)  
{  
    if (num % 2 == 0) evens.Add(num);  
}
```

```
// LINQ Approach  
var evens = numbers.Where(num => num % 2 == 0).ToList();
```

Common LINQ Methods

Most used LINQ methods in Method Syntax:

- **Select() – Project values**
- **Where() – Filter records**
- **OrderBy()/OrderByDescending() – Sort records**
- **GroupBy() – Group similar records**
- **Join() – Combine data from multiple sources**
- **Aggregate() – Perform calculations**

LINQ with Databases (Entity Framework)

- LINQ is used in Entity Framework (EF) to query databases efficiently.
- Example Query:



```
var students = db.Students.Where(s => s.Age > 18).ToList();
```

- **Benefits:**
- No need to write raw SQL
- Works seamlessly with ORM (Object Relational Mapping)

Performance Considerations

- Deferred Execution: Queries execute only when data is accessed.
- Avoid multiple `.ToList()` calls (loads entire data in memory).
- Use `AsEnumerable()` for in-memory filtering.

Real-World Use Cases

- Querying lists of students/employees
- Searching/filtering e-commerce products
- Processing JSON or XML data
- Reporting dashboards with aggregations
- Data migration and transformation

Conclusion

- LINQ simplifies data manipulation in C#.
- Works with arrays, lists, databases, XML, JSON.
- Query Syntax & Method Syntax – both have their use cases.
- Entity Framework + LINQ = Powerful Database Queries.
- Performance optimizations should be considered for large datasets.