LINQ Functions Summary

Friday, May 16, 2025 8:34 AM

Comprehensive summary of **LINQ (Language-Integrated Query)** in table format, categorized by different types of LINQ functions.

This table gives a quick and **comprehensive** summary of **LINQ functions** and **their usage**.

1. Filtering Operators

```
Where     var result = numbers.Where(n => n > 5);
OfType<T> var result = mixedList.OfType<int>();
```

2. Projection Operators

```
Select     var result = employees.Select(e => e.Name);
SelectMany var result = employees.SelectMany(e => e.Skills);
```

3. Sorting Operators

```
OrderBy           var result = employees.OrderBy(e => e.Name);
OrderByDescending var result = employees.OrderByDescending(e => e.Age);
ThenBy            var result = employees.OrderBy(e => e.Name).ThenBy(e => e.Age);
ThenByDescending  var result = employees.OrderBy(e => e.Name).ThenByDescending(e => e.Age);
Reverse           var result = numbers.Reverse();
```

4. Grouping Operators

```
GroupBy var result = employees.GroupBy(e => e.Department);
```

5. Joining Operators

```
Join      var result = employees.Join(departments, e => e.DeptId, d => d.Id, (e, d) => new { e.Name, d.Name });
GroupJoin var result = departments.GroupJoin(employees, d => d.Id, e => e.DeptId, (d, e) => new { d.Name, Employees = e });
```

6. Set Operators

```
Distinct  var result = numbers.Distinct();
Union     var result = list1.Union(list2);
Intersect var result = list1.Intersect(list2);
Except    var result = list1.Except(list2);
```

7. Aggregation Operators

```
Count     var count = numbers.Count();
Sum       var sum = numbers.Sum();
Average   var avg = numbers.Average();
Min       var min = numbers.Min();
Max       var max = numbers.Max();
Aggregate var result = numbers.Aggregate((a, b) => a + b);
```

8. Quantifier Operators

```
Any      var result = employees.Any(e => e.Age > 40);
All      var result = employees.All(e => e.Age > 18);
Contains var result = numbers.Contains(10);
```

9. Element Operators

```
First              var result = employees.First();
FirstOrDefault     var result = employees.FirstOrDefault();
Last               var result = employees.Last();
LastOrDefault      var result = employees.LastOrDefault();
Single             var result = employees.Single(e => e.Id == 1);
SingleOrDefault    var result = employees.SingleOrDefault(e => e.Id == 1);
```

```
ElementAt          var result = employees.ElementAt(2);
ElementAtOrDefault var result = employees.ElementAtOrDefault(2);
DefaultIfEmpty     var result = numbers.DefaultIfEmpty(0);
```

10. Conversion Operators

```
ToList       var result = employees.ToList();
ToArray      var result = employees.ToArray();
ToDictionary var result = employees.ToDictionary(e => e.Id, e => e.Name);
Cast<T>      var result = mixedList.Cast<int>();
AsEnumerable var result = employees.AsEnumerable();
AsQueryable  var result = employees.AsQueryable();
```