Where

Where Operator

✓ **Where** comes under Filtering operators.

✓ Where is used to filter specific data from a data source based on a condition.

✓ Where always expects a condition

✓ Condition can be written using following symbols –

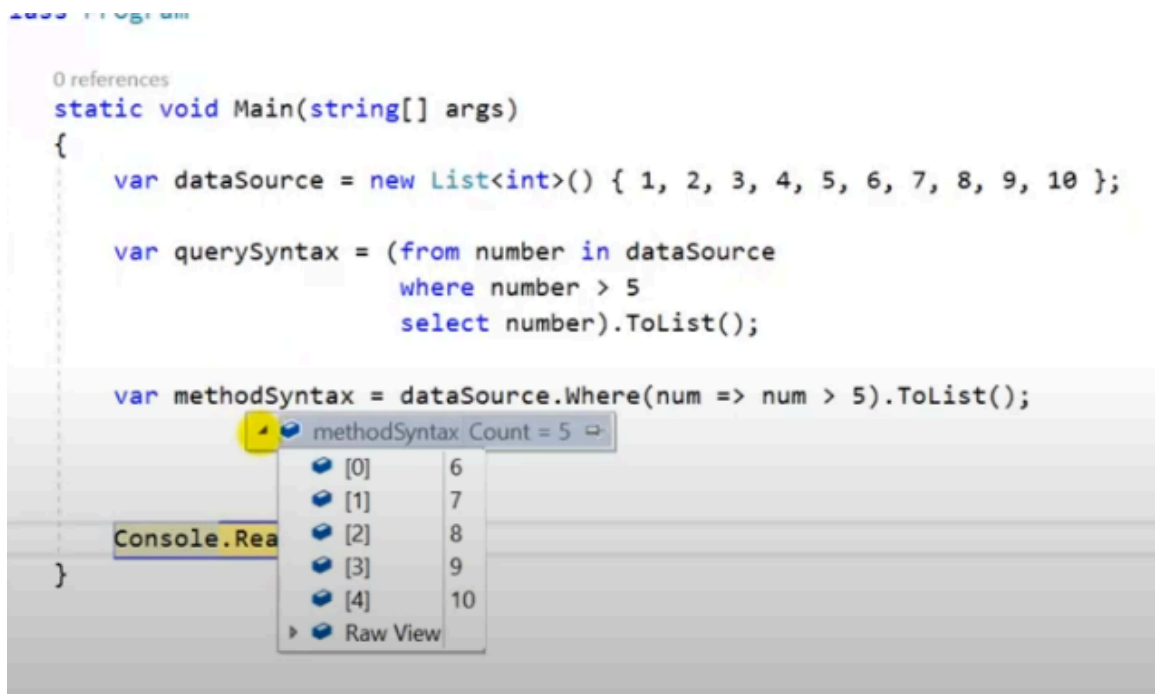|			
✓ ==	✓ >=	✓ <=	✓ &&
✓ >	✓ <	✓ ||	✓ Etc.

Example

```
0 references
static void Main(string[] args)
{
    var dataSource = new List<int>() { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

    var querySyntax = (from number in dataSource
                       where number > 5
                       select number).ToList();

    var methodSyntax = dataSource.Where(num => num > 5).ToList();
            methodSyntax  Count = 5
                [0]        6
                [1]        7
    Console.Rea [2]        8
                [3]        9
}               [4]        10
              Raw View
```

Multiple conditions

```
static void Main(string[] args)
{
    var dataSource = new List<int>() { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

    var querySyntax = (from number in dataSource
                       where number <= 5 || number > 9
                       select number).ToList();

    var methodSyntax = dataSource.Where(num => num <= 5 || num > 9).ToList();



    Console.ReadLine();
}
```

IN STRING

```
var dataSource = new List<string>() { "Tom", "Harry", "Adam", "Ponting", "Sachin" };

var querySyntax = (from str in dataSource
                   where str == "Tom" || str.Length > 5
                   select str).ToList();

var methodSyntax = dataSource.Where(str => str.Length > 5 || str == "Tom").ToList();
```
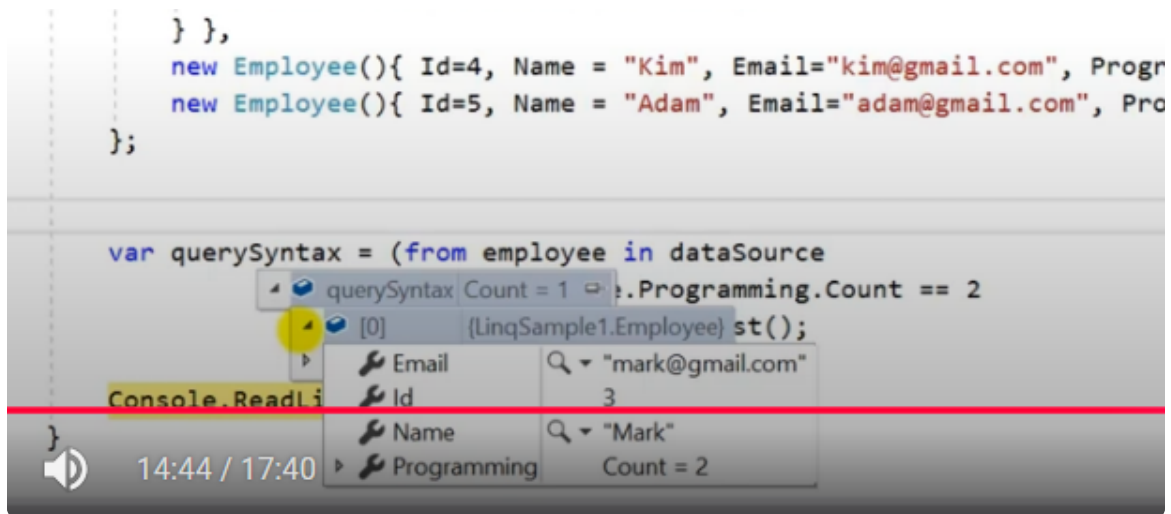
Using Count

```
        new Techs() { Technology = "MVC"},
        new Techs() { Technology = "SQL"}
    } },
    new Employee(){ Id=2, Name = "John", Email="john@gmail.com", Programming =
    new List<Techs>{
        new Techs() { Technology = "C#"},
        new Techs() { Technology = "VB"},
        new Techs() { Technology = "SQL"}
    } },
    new Employee(){ Id=3, Name = "Mark", Email="mark@gmail.com", Programming =
    new List<Techs>{
        new Techs() { Technology = "LINQ"},
        new Techs() { Technology = "MVC"}
    } },
    new Employee(){ Id=4, Name = "Kim", Email="kim@gmail.com", Programming = new List<Techs>()} ,
    new Employee(){ Id=5, Name = "Adam", Email="adam@gmail.com", Programming = new List<Techs>()}
};


var querySyntax = (from employee in dataSource
                   where employee.Programming.Count == 0
                   select employee).ToList();
```

```
      } },
      new Employee(){ Id=4, Name = "Kim", Email="kim@gmail.com", Progr
      new Employee(){ Id=5, Name = "Adam", Email="adam@gmail.com", Pro
};

var querySyntax = (from employee in dataSource
          querySyntax Count = 1      .Programming.Count == 2
          [0]         {LinqSample1.Employee} st();
            Email              "mark@gmail.com"
Console.ReadLi  Id              3
                Name           "Mark"
              Programming      Count = 2
}
```

14:44 / 17:40

Muptiple queries

```
var querySyntax = (from employee in dataSource
                   where employee.Programming.Count == 2
                   && employee.Id == 4
                   select employee).ToList();

var methodSyntax = dataSource.Where(employee => employee.Programming.Count > 3).ToList();
          methodSyntax Count = 1
Console.Re   [0]         {LinqSample1.Employee}
}            Email        "tom@gmail.com"
             Id           1
16:35 / 17:40 Name        "Tom"
             Programming  Count = 5
```