

0 references

```
static void Main(string[] args)
{

    // Let us create a Sample Array

    var words = new string[]
    {
        "flower", "elephant", "sugar", "tree", "waffer"
    };

    // Write a Linq Query to Fetch all Elements from this
    // Array

    // Syntax:

    // from <Variable> in <Array> select <Variable>

    var result = from element in words select element;

    // write a For Each Loop to Print All Elements

    foreach(var r in result)
    {
        Console.WriteLine(r);
    }

}
```

0 references

```
static void Main(string[] args)
{

    // Let us create a Sample Array

    var words = new string[]
    {
        "flower", "elephant", "sugar", "tree", "waffer"
    };

    // Let us Write a Linq Query to Filter Elements
    // that contains a

    var result = from element in words
        where element.Contains("a")
        select element;

    foreach(var r in result)
    {
        Console.WriteLine(r);
    }
}
```

Microsoft Visual Studio

```
elephant
sugar
waffer

C:\Users\Rajen\PC\source\repos\LINQDemo_s\L
INQDemo_s\bin\Debug\net6.0\LINQDemo_s.exe (pr
ocess 21252) exited with code 0.
Press any key to close this window . . .
```

0 references

```
static void Main(string[] args)
{

    // Let us create a Sample Array

    var words = new string[]
    {
        "flower", "elephant", "sugar", "tree", "waffer"
    };

    // Short and Alternate Method

    var result = words.Where(element => element.Contains("a"));

    // Let us Write a Linq Query to Filter Elements
    // that contains a

    /* var result = from element in words
        where element.Contains("a")
        select element; */

    foreach(var r in result)
    {
        Console.WriteLine(r);
    }
}
```

Microsoft Visual Studio

```
elephant
sugar
waffer

C:\Users\Rajen\PC\source\repos\LINQDemo_s\L
INQDemo_s\bin\Debug\net6.0\LINQDemo_s.exe (pr
ocess 4668) exited with code 0.
Press any key to close this window . . .
```

```
static void Main(string[] args)
{

    // Let us create a Sample Array

    var words = new string[]
    {
        "flower", "elephant", "sugar", "tree", "waffer"
    };

    // Here are Some Helper Methods in Linq that Help you Find the
    // required Information QUICKLY!!!

    // fetchign element at 1 Index
    Console.WriteLine( words.ElementAt(1) );

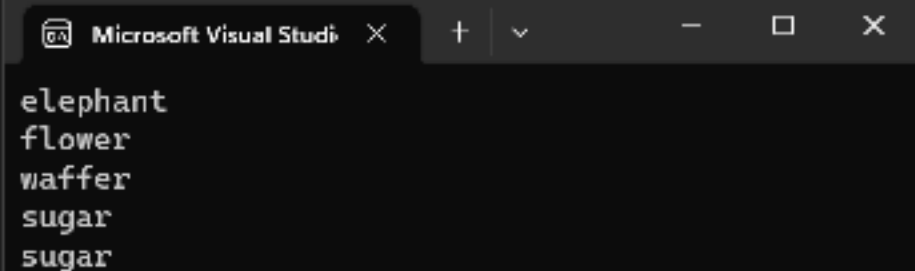
    // fetching First Element
    Console.WriteLine( words.First() );

    // Fetching Last Element
    Console.WriteLine( words.Last() );

    // Fetching First Element Whose Length matches with 5
    Console.WriteLine( words.First( element => element.Length == 5 ) );

    // Fetching last Element Whose Length matches with 5
    Console.WriteLine(words.Last(element => element.Length == 5));

}
```



Microsoft Visual Studi x + - □ x

```
elephant
flower
waffer
sugar
sugar
```


0 references

```
static void Main(string[] args)
{
    // Prepend vs Append

    int[] luckyNumbers = { 47, 91, 88 };

    // prepend is used to add element at the
    // beginning of Array

    var result = luckyNumbers.Prepend(60);

    // append() is used to add element at the
    // end of array

    var result2 = result.Append(33);

    // .join() this Converts Array to String

    Console.WriteLine( string.Join("-",result2) );

    // in Join Method ("-") represents the
    // symbol to be added when merging
    // array elements. it can be any.

    // Eleminating Duplicates using .distinct()

    int[] dups = { 1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, };

    var uniques = dups.Distinct();

    foreach (var u in uniques)
    {
        Console.WriteLine(u);
    }
}
```

Microsoft Visual Studi x

60-47-91-88-33
1
2
3
4

C:\Users\Rajan's PC\source\repos\LINQDeno_s\L
INQDeno_s\bin\Debug\net6.0\LINQDeno_s.exe (pr
ocess 21876) exited with code 0.
Press any key to close this window . . .

0 references

```
static void Main(string[] args)
{
    // using .Select() Method
    // select Method is used to Compute Each element of Given Array and
    // Produce a New Resultset.

    // here is an Example

    int[] numbers = { 1, 2, 3, 4, 5 };

    // finding the Square of Each Number.

    var result = numbers.Select(num => num * num);

    foreach( var i in result)
    {
        Console.WriteLine(i);
    }
}
```


11 references

class User

{

public int id;

public string name;

public string city;

public string dateofbirth;

}

0 references

internal class Program

{

0 references

static void Main(string[] args)

{

// Working with Objects Array using LINQ

User[] users = {

new User { id = 1, name="john", city="london", dateofbirth="2001-05-21" },

new User { id = 2, name="windy", city="singapore", dateofbirth="2002-06-23" },

new User { id = 3, name="barbara", city="hongkong", dateofbirth="2003-08-7" },

new User { id = 4, name="nick", city="hongkong", dateofbirth="2001-04-9" },

new User { id = 5, name="paul", city="singapore", dateofbirth="2004-03-23" },

new User { id = 6, name="andika", city="singapore", dateofbirth="2001-12-23" },

new User { id = 7, name="herman", city="london", dateofbirth="2001-07-21" },

new User { id = 8, name="sadiq", city="london", dateofbirth="2003-4-21" },

new User { id = 9, name="casselyn", city="hongkong", dateofbirth="2004-05-21" },

new User { id = 10, name="miki", city="singapore", dateofbirth="2005-08-21" }

};

// Linq Query to Find Hongkong Users Only.

var result = from usr in users where usr.city == "hongkong" select usr;

foreach (var usr in result)

{

Console.WriteLine(usr.name + " " + usr.city);

}

}

}

0 references

```
static void Main(string[] args)
{
    // Using .split() function to Convert a String into Array

    String wish = "One #Day I'll Make the #Onions #CRY";

    // Objective: to extract HashTags

    String[] words = wish.Split(' ');

    var result = words.Where(w => w.StartsWith("#"));

    foreach (var word in result)
    {
        Console.WriteLine(word);
    }
}
```



```
// Various Statistical Methods
// such as / Count() / Max() / Min() / Sorting / Sum / Average

var numbers = new List<int> { 6, 2, -3, 4, -5, 9, 7, 8 };

int length = numbers.Count();
Console.WriteLine($"Total Elements {length}");

// Count Even Numbers

int length2 = numbers.Count(e => e % 2 == 0);
Console.WriteLine($"Even Numbers Length {length2}");

// Find Sum

int total = numbers.Sum();
Console.WriteLine($"Sum {total}");

// Find Sum of Even Numbers
int evenSum = numbers.Sum(e => e % 2 == 0 ? e : 0);
Console.WriteLine($"Even Numbers Sum {evenSum}");

// Find Average
double mean = numbers.Average();
Console.WriteLine($"Even Numbers Average {mean}");

// MAX
Console.WriteLine($"Largest Value {numbers.Max()}");

// MIN
Console.WriteLine($"Smallest Value {numbers.Min()}");

// sorting
var result = from num in numbers orderby num ascending select num;
foreach (int num in result) { Console.Write(num + " "); }
```


6 references

class Employee

```
{  
    public string name;  
    public string gender;  
}
```

0 references

internal class Program

```
{  
    0 references  
    static void Main(string[] args)  
    {
```

// Understanding Grouping

```
Employee[] employees = {  
    new Employee{ name = "windy", gender="Female"},  
    new Employee{ name = "james", gender="Male"},  
    new Employee{ name = "casselyn", gender="Female"},  
    new Employee{ name = "armanta", gender="Female"},  
    new Employee{ name = "jonSeto", gender="Male"},  
};
```

```
var empGroups = from emp in employees group emp by emp.gender;
```

// this loop iterates through GROUP

```
foreach (var grp in empGroups)  
{
```

```
    Console.WriteLine("Group -> " + grp.Key);
```

// this loop iterates through each Group Elements.

```
foreach (var e in grp)  
{  
    Console.WriteLine(e.name + " " + e.gender);  
}
```

```
}  
}
```

Microsoft Visual Studio

```
Group -> Female  
windy Female  
casselyn Female  
armanta Female  
Group -> Male  
james Male  
jonSeto Male  
  
C:\Users\Rajan's PC\source\repos\LINQDemo_s\L  
INQDemo_s\bin\Debug\net6.0\LINQDemo_s.exe (pr  
ocess 17936) exited with code 0.  
Press any key to close this window . . .
```