**JavaScript Array Filter**

24191% *of* 997 *of* 5,282[jhoffner](https://www.codewars.com/users/jhoffner)

C#

* [Train Again](https://www.codewars.com/kata/javascript-array-filter/train/csharp)
* [Next Kata](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/javascript-array-filter/solutions/csharp)

[Discourse (20)](https://www.codewars.com/kata/javascript-array-filter/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Starting with .NET Framework 3.5, C# supports a Where (in the System.Linq namespace) method which allows a user to filter arrays based on a predicate. Use the Where method to complete the function given.

Enumerable.Where documentation: [https://msdn.microsoft.com/en-us/library/bb534803(v=vs.110).aspx](https://msdn.microsoft.com/en-us/library/bb534803%28v=vs.110%29.aspx)

The solution would work like the following:

Kata.GetEvenNumbers(new int[] {2, 4, 5, 6}) => new int[] {2, 4, 6}

<https://www.codewars.com/kata/javascript-array-filter/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

public static int[] GetEvenNumbers(int[] numbers)

{

// filter out the odd numbers

List<int> pares = new List<int>();

for (int i = 0; i < numbers.Length; i++)

{

if (numbers[i] % 2 == 0)

{

pares.Add(numbers[i]);

}

}

return pares.ToArray();

}

public class Kata

{

public static int[] GetEvenNumbers(int[] numbers)

{

return numbers.Where(a => a % 2 == 0).ToArray();

}

}

static void Main(string[] args)

{

int[] arr = new int[] { 2, 4, 0, -1, -3 };

Console.ReadLine();

}

}

}