**List Filtering**

1481687% *of* 1,83025 *of* 17,518[cmgerber](https://www.codewars.com/users/cmgerber)

C#

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

Details

[Solutions](https://www.codewars.com/kata/list-filtering/solutions/csharp)

[Discourse (65)](https://www.codewars.com/kata/list-filtering/discuss/csharp)

* Add to Collection
* |
* Share this kata:

In this kata you will create a function that takes a list of non-negative integers and strings and returns a new list with the strings filtered out.

**Example**

ListFilterer.GetIntegersFromList(new List<object>(){1, 2, "a", "b"}) => {1, 2}

ListFilterer.GetIntegersFromList(new List<object>(){1, 2, "a", "b", 0, 15}) => {1, 2, 0, 15}

ListFilterer.GetIntegersFromList(new List<object>(){1, 2, "a", "b", "aasf", "1", "123", 231}) => {1, 2, 231}

Fundamentals

Lists

Data Structures

Filtering

Algorithms

<https://www.codewars.com/kata/list-filtering/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

public static IEnumerable<int> GetIntegersFromList(List<object> listOfItems)

{

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

for (int i = 0; i < listOfItems.Count; i++)

{

if (listOfItems[i] is int)

{

s.Add((int)listOfItems[i]);

}

}

return (IEnumerable<int>)s;

}

public static IEnumerable<int> GetIntegersFromList(List<object> listOfItems)

{

return listOfItems.OfType<int>().Cast<int>();

}

static void Main(string[] args)

{

List<object> lista = new List<object>() { 1, 2, "a", "b" };

IEnumerable<int> res = GetIntegersFromList(lista);

foreach (int elem in res)

{

Console.Write(elem + " ");

}

Console.ReadLine();

}

}

}