**7 kyu**

**Every nth array element. (Basic)**

19389% of 9647 of490[Insti](https://www.codewars.com/users/Insti)

C#

* [TRAIN AGAIN](https://www.codewars.com/kata/every-nth-array-element-basic/train/csharp)
* [NEXT KATA](https://www.codewars.com/trainer/csharp)

Details

[Solutions](https://www.codewars.com/kata/every-nth-array-element-basic/solutions/csharp)

[Forks (2)](https://www.codewars.com/kata/every-nth-array-element-basic/forks/csharp)

[Discourse (27)](https://www.codewars.com/kata/every-nth-array-element-basic/discuss/csharp)

* Add to Collection
* |
* Share this kata:

Create a method (**JS**: function) every which returns every nth element of an array.

**Usage**

With no arguments, array.every it returns every element of the array.  
With one argument, array.every(interval) it returns every intervalth element.  
With two arguments, array.every(interval, start\_index) it returns every intervalth element starting at index start\_index

**Note**: due to fact JS translation ask for a function instead of an Array method the above uses will be :

javascript: ruby:

every(array) array.every

every(array, interval) array.every(interval)

every(array, interval, start\_index) array.every(interval, start\_index)

**Examples**

/\* C# can work either as "Kata.Every" or "array.Every" due to the nature of extension functions. \*/

new int []{0,1,2,3,4}.Every()); // [0,1,2,3,4]

Kata.Every(new int []{0,1,2,3,4}, 5, 1)); // [1]

**Notes**

Test cases:

interval will always be a positive integer (but may be larger than the size of the array).  
start\_index will always be within the bounds of the array.

Once you have completed this kata, try the **advanced version** (<http://www.codewars.com/kata/every-nth-array-element-advanced>) which allows negative intervals and unbounded start\_indexes

**Translator notes**

Ruby is the original language for this kata.

<https://www.codewars.com/kata/every-nth-array-element-basic/csharp>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

public static class Kata

{

public static IEnumerable<T> Every<T>(this IEnumerable<T> arr, int interval, int start)

{

List<T> ret = new List<T>();

List<T> tmp = arr.ToList();

for (int i = start; i < tmp.Count; i += interval)

{

ret.Add(tmp[i]);

}

return ret;

}

public static IEnumerable<T> Every<T>(this IEnumerable<T> arr, int interval)

{

List<T> ret = new List<T>();

List<T> tmp = arr.ToList();

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

{

ret.Add(tmp[i]);

}

return ret;

}

public static IEnumerable<T> Every<T>(this IEnumerable<T> arr)

{

return arr;

}

}

class Program

{

static void Main(string[] args)

{

Console.ReadLine();

}

}

}