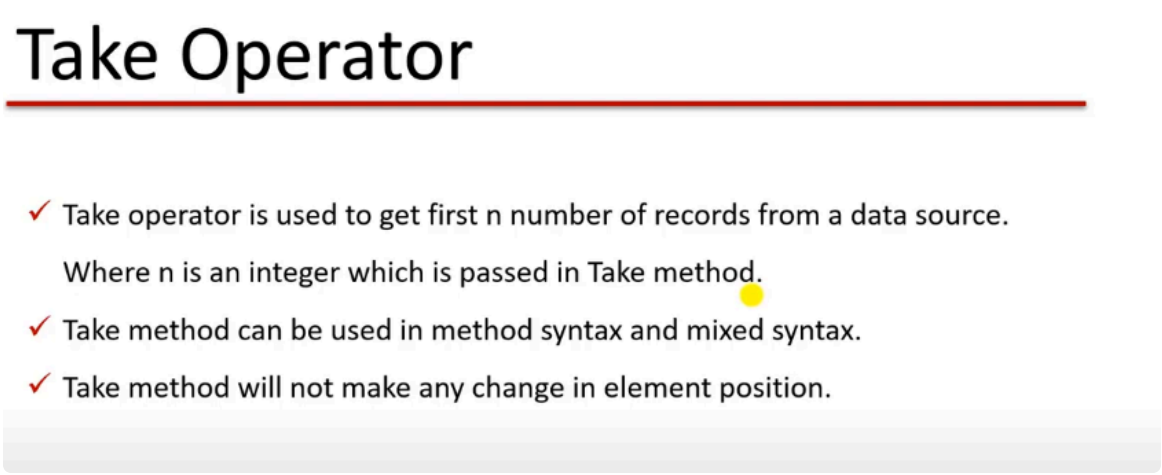
Patition methods

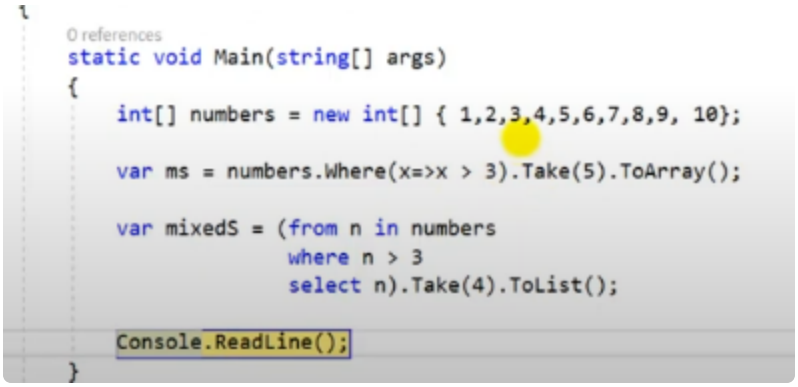
Take Operator



Example

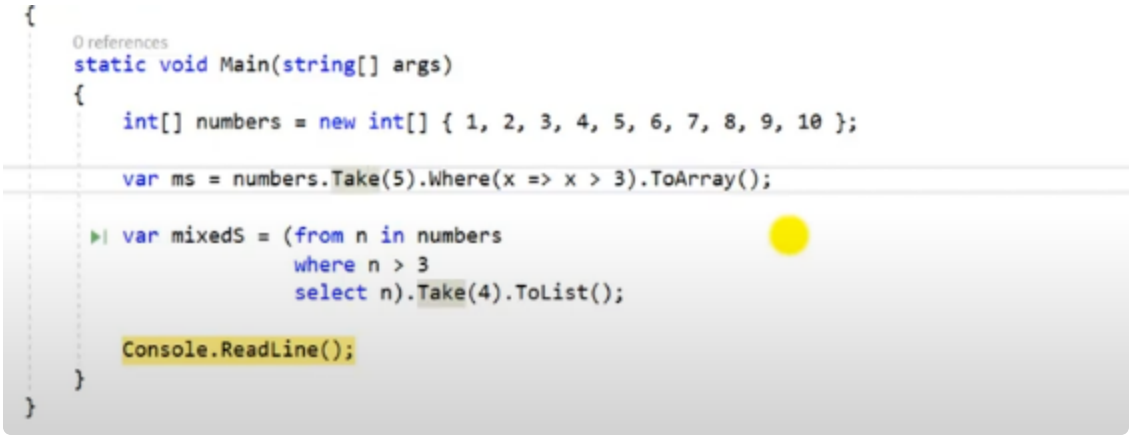
```
ıss Program

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

        var ms = numbers.Take(5).ToArray();

        var mixedS = (from n in numbers select n).Take(4).ToList();

        Console.ReadLine();
    }
```

To filter and take from it



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

    var ms = numbers.Where(x=>x > 3).Take(5).ToArray();

    var mixedS = (from n in numbers
                  where n > 3
                  select n).Take(4).ToList();

    Console.ReadLine();
}
```

We can also use the where after the take remember it will perform the operation on the take data

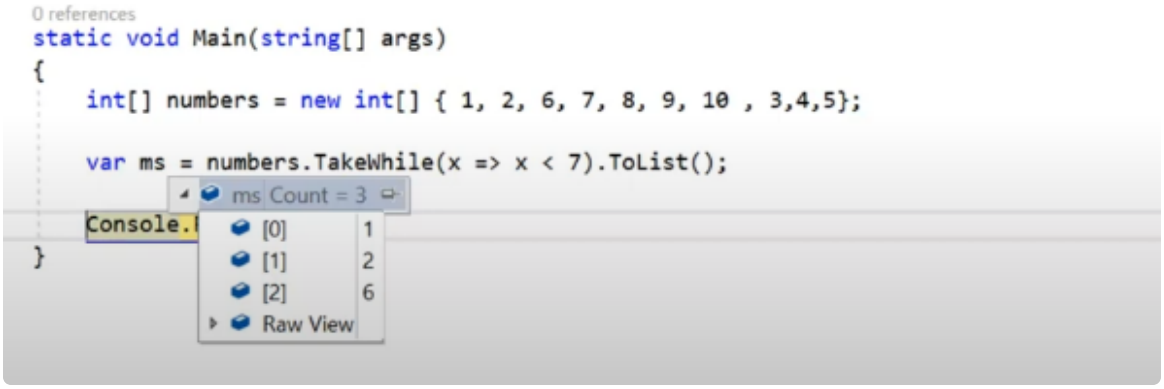


```
{
    0 references
    static void Main(string[] args)
    {
        int[] numbers = new int[] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

        var ms = numbers.Take(5).Where(x => x > 3).ToArray();

        var mixedS = (from n in numbers
                      where n > 3
                      select n).Take(4).ToList();

        Console.ReadLine();
    }
}
```

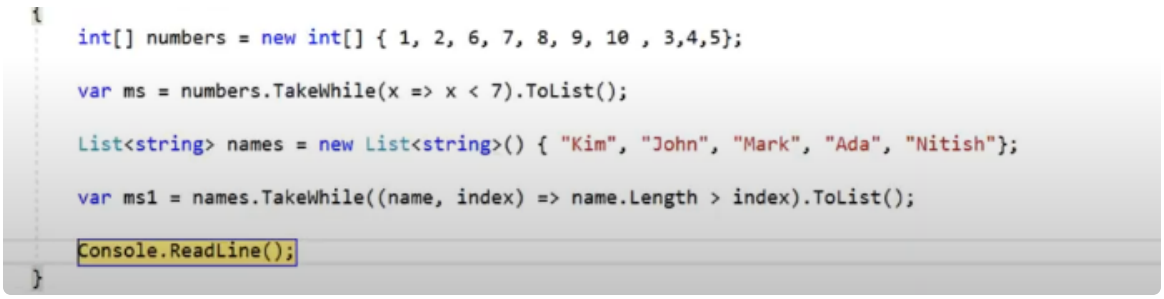
Take While

TakeWhile Operator

- ✓ TakeWhile operator is used to get all records from a data source until a specified condition is true.
- ✓ Once the condition is failed TakeWhile will not validate rest elements Even if the condition is true for remaining elements .
- ✓ TakeWhile method can be used in method syntax and mixed syntax.
- ✓ TakeWhile method will not make any change in element position.



Another example to check whether the name length is greater than the current index



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

    var ms = (from n in numbers select n).TakeWhile(x => x < 7).ToList();

    List<string> names = new List<string>() { "Kim", "John", "Mark", "Ada", "Nitish"};

    var ms1 = (from n in names select n).TakeWhile((name, index) => name.Length > index).ToList();

    Console.ReadLine();
}
```

Skip Operator

- ✓ Skip operator is used to skip first n number of records from a data source and select remaining elements as an output.
  where n is an integer which is passed in Skip method.

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

    List<string> names = new List<string>() { "Kim", "John", "Mark", "Adam", "Nitish"};


    var ms = numbers.Skip(3).ToArray();
    Console.R                          ();
}
```

ms {int[7]}
[0] 7
[1] 8
[2] 9
[3] 10
[4] 3
[5] 4
[6] 5

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

    List<string> names = new List<string>() { "Kim", "John", "Mark", "Adam", "Nitish" };


    var ms = numbers.Where(x => x > 4).Skip(3).ToArray();

    var mixedS = (from n in names select n).Skip(2).ToList();

    Console.ReadLine();
}
```

SkipWhile Operator

- ✓ SkipWhile operator is used to skip all records from a data source until a condition is true and select remaining elements as an output.

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

    List<string> names = new List<string>() { "Kim", "John", "Ma", "Adam", "Nitish" };


    var ms = numbers.SkipWhile(num => num < 6).ToArray();

    var mixedS = (from num in numbers select num).SkipWhile(x => x < 5).ToArray();

    var ms1 = names.SkipWhile((value, index) => value.Length < index).ToList();
    Console.Re
}
```

ms1 Count = 5

[0]	"Kim"
[1]	"John"
[2]	"Ma"
[3]	"Adam"
[4]	"Nitish"