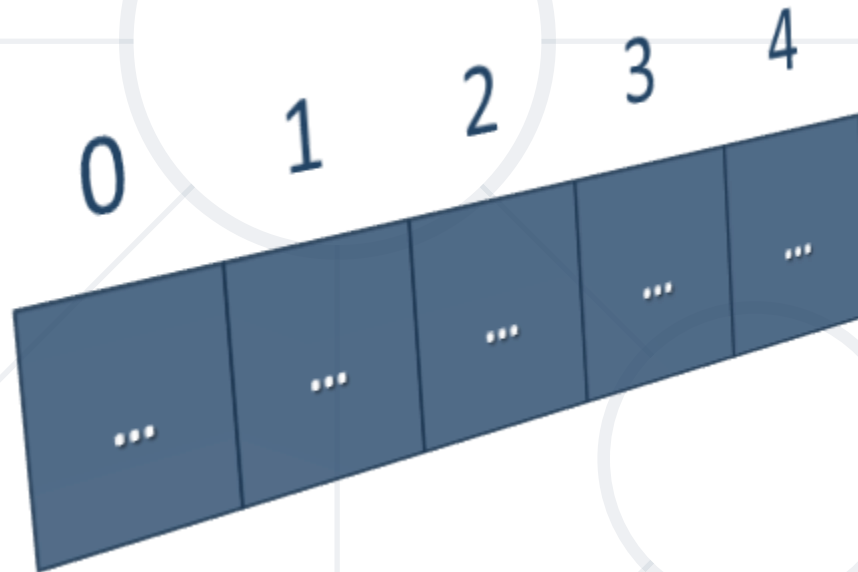


Масиви

Съвкупност от елементи с определена дължина



СофтУни

Преподавателски екип



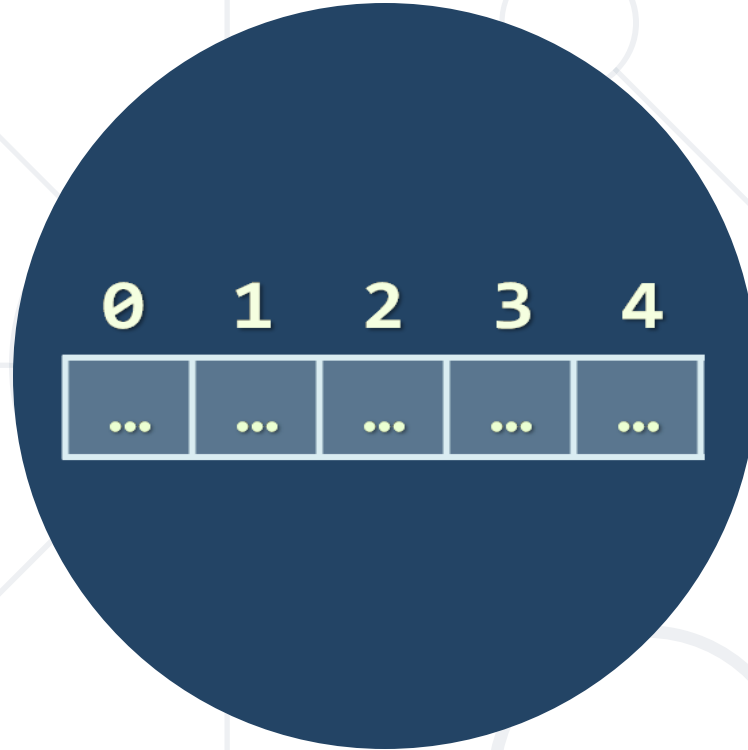
SoftUni

Софтуерен университет

<https://softuni.bg>

1. **Масиви**
2. **Четене** на масиви от конзолата
3. **Foreach** цикъл





Работа с елементи на масива

Какво означава масив?

- В програмирането, **масивът** е **последователност от елементи**



- Елементите са номерирани от **0** до **дължината-1**
- Елементите са от **един и същ тип данни** (например числа)
- Масивите имат **фиксирана дължина** (**Array.Length**) и не може да се променя

Създаване на масив

- Изполвайте ключовата дума **new**
 - Думата се използва за създаване на масива и инициализиране на елементите с тяхната стойност по подразбиране

- Създаване на **масив** от 10 **числа**:

```
int[] numbers = new int[10];
```

Всички елементи първоначално са **0**

- По същия начин можем да създадем масив от **НИЗОВЕ**:

```
string[] names = new string[10];
```

Всички елементи първоначално са **null**



- **Слагане на стойност** на елементите от масива
 - **Дължината** представлява броя на елементите в масива

```
for (int i = 0; i < numbers.Length; i++)  
    numbers[i] = 1;
```

- **Достъп** до елементите на масива чрез индекс
 - Операторът **[]** осъществява достъп до елементите на масива

```
numbers[5] = numbers[2] + numbers[7];  
numbers[10] = 1; // IndexOutOfRangeException
```

- Дните от седмицата могат да бъдат съхранявани в **массив от низове**:

```
string[] days = {  
    "Monday",  
    "Tuesday",  
    "Wednesday",  
    "Thursday",  
    "Friday",  
    "Saturday",  
    "Sunday"  
};
```



Operator	Notation in C#
days[0]	Monday
days[1]	Tuesday
days[2]	Wednesday
days[3]	Thursday
days[4]	Friday
days[5]	Saturday
days[6]	Sunday

Задача: Ден от седмицата

- Въведете **ден от седмицата** [1...7] и отпечатайте **името на деня** (на английски) или **"Invalid day!"**

Name	Value	Type
days	{string[7]}	string[]
• [0]	"Monday"	string
• [1]	"Tuesday"	string
• [2]	"Wednesday"	string
• [3]	"Thursday"	string
• [4]	"Friday"	string
• [5]	"Saturday"	string
• [6]	"Sunday"	string

Решение: Ден от седмицата

```
string[] days = { "Monday", "Tuesday", "Wednesday",  
"Thursday", "Friday", "Saturday", "Sunday" };
```

```
int day = int.Parse(Console.ReadLine());
```

```
if (day >= 1 && day <= 7)
```

```
    Console.WriteLine(days[day - 1]);
```

```
else
```

```
    Console.WriteLine("Invalid day!");
```

Първият ден в масива е на индекс 0, не на 1.



Използване на цикъл или `String.Split()`

- Първо четем **дължината** на масива от конзолата:

```
int n = int.Parse(Console.ReadLine());
```

- След това създаваме масив с дължина **n** и четем неговите **елементи**:

```
int[] arr = new int[n];  
for (int i = 0; i < n; i++)  
{  
    arr[i] = int.Parse(Console.ReadLine());  
}
```

- Масивите могат да бъдат **четени** от **един ред** с **разделени стойности**

```
2 8 30 25 40 72 -2 44 56
```

```
string values = Console.ReadLine();  
string[] items = values.Split();  
int[] arr = new int[items.Length];  
  
for (int i = 0; i < items.Length; i++)  
    arr[i] = int.Parse(items[i]);
```

Split() разделя
чрез място в
string[]

- Четене на масив от **числа**:

using System.Linq;

```
var inputLine = Console.ReadLine();  
string[] items = inputLine.Split(', ');  
int[] arr = items.Select(int.Parse).ToArray();
```

```
int[] arr = Console.ReadLine().Split(' ', '  
    .Select(int.Parse).ToArray();
```

Или
пократко

- За да **принтираме** всички елементи на масива, може да използваме **for-цикъл**
- Разделяме елементите с интервал или нов ред

```
string[] arr = {"one", "two"};
// == new string [2] {"one", "two"};
// Обработване на елементите на масива
for (int index = 0; index < arr.Length; index++)
{
    //Принтираме всеки елемент на нов ред
    Console.WriteLine("arr[{0}] = {1}", index, arr[index]);
}
```

Задача: Отпечатване на числа в обратен ред

- Прочетете масив от числа (**n** реда с числа), **обърнете** го и принтирайте елементите на нов ред, разделени с интервал:

3
10
20
30



30 20 10

4
-1
20
99
5



5 99 20 -1

Решение: Отпечатване на числа в обратен ред

```
// Прочитане на масива (n на брой реда от числа)
var n = int.Parse(Console.ReadLine());
var arr = new int[n];
for (int i = 0; i < n; i++) {
    arr[i] = int.Parse(Console.ReadLine()); }
// Отпечатване на масива от последния до първия елемент
for (int i = n-1; i >= 0; i--) {
    Console.Write(arr[i] + " "); }
Console.WriteLine();
```


Задача: Закръглени числа

- Прочетете **масив от реални числа** (разделени с интервал), закръглете ги по начина "**away from 0**" отпечатайте изхода както е в примера:

0.9 1.5 2.4 2.5 3.14



0.9 => 1

1.5 => 2

2.4 => 2

2.5 => 3

3.14 => 3

-5.01 -1.599 -2.5 -1.50 0



-5.01 => -5

-1.599 => -2

-2.5 => -3

-1.50 => -2

0 => 0

- **Rounding** закръгля всяка стойност до най-близкото число

```
double[] nums = Console.ReadLine().Split()  
    .Select(double.Parse).ToArray();  
int[] roundedNums = new int[nums.Length];  
for (int i = 0; i < nums.Length; i++) {  
    roundedNums[i] = (int)Math  
        .Round(nums[i], MidpointRounding.AwayFromZero); }  
// TODO: Принтирайте всяко число
```

2.5 => 3

- Чрез for-цикъл:

```
int[] arr = { 10, 20, 30, 40, 50};  
for (int i = 0; i < arr.Length; i++) {  
    Console.WriteLine(arr[i]); }
```

- Чрез **string.Join(разделител, масив)**:

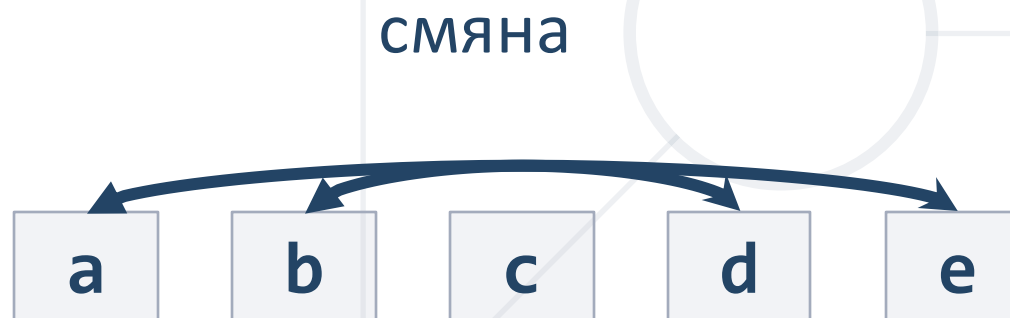
```
int[] arr = { 1, 2, 3 };  
Console.WriteLine(string.Join(", ", arr)); // 1, 2, 3  
string[] strings = { "one", "two" };  
Console.WriteLine(string.Join(" - ", strings)); // едно - две
```

Задача: Обърнат масив от низове

- Прочетете **масив от низове** (стойностите са разделени с интервал), **обърнете го** и **принтирайте** елементите:

a b c d e ➡ e d c b a -1 hi ho w ➡ w ho hi -1

- Обръщане на елементите на масив:



```
var items = Console.ReadLine().Split(' ').ToArray();
for (int i = 0; i < items.Length / 2; i++)
{
    var oldElement = items[i];
    items[i] = items[items.Length - 1 - i];
    items[items.Length - 1 - i] = oldElement;
}


Console.WriteLine(string.Join(" ", items));
```




Обхождане на колекции

Foreach цикъл

- Преминава през **всички елементи** в колекцията
- **Няма** достъп до сегашния индекс
- **Read-only**

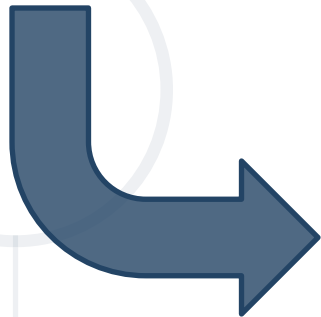


```
foreach (var item in collection)
{
    // Използване на стойността тук
}
```



Принтиране на масив чрез Foreach

```
int[] numbers = { 1, 2, 3, 4, 5 };  
foreach (int number in numbers)  
{  
    Console.Write($"{number} ");  
}
```



1 2 3 4 5

- Масивите представляват **съвкупност** от елементи
 - Елементите са номерирани от **0** до **дължина-1**
- Създаване на масив: **new[]**
- Достъп до елементите чрез **индекс**
- Отпичатване на елементите: **string.Join()**

Въпроси?



SoftUni



Software
University



SoftUni
Svetlina



SoftUni
Creative



SoftUni
Digital



SoftUni
Foundation



SoftUni
Kids

- Този курс (презентации, примери, демонстрационен код, упражнения, домашни, видео и други активи) представлява **защитено авторско съдържание**
- Нерегламентирано копиране, разпространение или използване е незаконно
- © СофтУни – <https://softuni.org>
- © Софтуерен университет – <https://softuni.bg>

